PART II
RESEARCH PAPERS

2. METRICS FOR INTELLIGENT SYSTEMS

2.1 Performance Metrics for Intelligent Systems
J.M. Evans, E.R. Messina, NIST, USA

2.2. The Search for Metrics of Intelligence -- A Critical View
L.A. Zadeh, University of California at Berkeley, USA

2.3 Measure of System Intelligence: An Engineering Perspective
S. Lee, SCS, SAIT, Korea, W.-C. Bang, and Z.Z. Bien, KAIST, Korea

2.4 Formal Specification of Performance Metrics for Intelligent Systems
Y. Zhang, Xerox, USA, A.K. Mackworth, University of BC, Canada

2.5 Metrics for Intelligence: the Perspective from Software Agents
PRELIMINARY NOTES
L. Pouchard, Oak Ridge National Lab, USA

2.6 Minimal Representation Size Metrics for Intelligent Robotic Systems
A. Sanderson, NSF, USA

2.7 Metrics for System Autonomy. Part I: Metrics Definition
A. Yavnai, RAFAEL, Israel

2.8 In Defense of the Additive Form for Evaluating the Multidimensional Vector
L.M. Polyakov, Globe Institute of Technology, USA

2.9 A Method for Evaluating the IQ of Intelligent Systems,
R. Finkelstein, Robotic Technology, Inc., USA

2.10 Machine Intelligence Ranking
P.P. Wang, Duke University, USA

2.11 Survivability and Competence as Measures of Intelligent Systems
R. Simmons, Carnegie-Mellon University, USA

2.12 Two measures for measuring the `intelligence' of human-interactive
robots in contests and in the real world: expressiveness and perceptiveness
I.R. Nourbakhsh, Carnegie-Mellon University, USA

Performance Metrics for Intelligent Systems

John M. Evans and Elena R. Messina
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

ABSTRACT

Research into intelligent systems and intelligent control is burgeoning. However, there is no consensus on how to define or measure an intelligent system. This lack of rigor hinders the ability to measure progress in the field and to compare different systems' capabilities. We discuss some of the challenges and issues in defining performance metrics for intelligent systems and issue a call to action to participants in the Performance Metrics for Intelligent Systems Workshop to define practical metrics that will advance the state of the art and practice.

KEYWORDS: *performance metrics, intelligent systems, intelligent control*

1. INTRODUCTION

Intelligent systems are increasingly being identified as solutions to many advanced applications in manufacturing, defense, and other domains. Industry workshops [4] and roadmaps [3] specifically call for intelligent control or intelligent systems to address needs such as

- Adaptive, reconfigurable manufacturing equipment and processes
- Self-optimizing, science-based control of manufacturing unit processes
- "First part correct," that is, the ability to design and manufacture a product correctly, the first time and every time
- Self-diagnosing and self-maintaining systems
- Tool wear and breakage monitoring

Government agencies are basing major programs on intelligent capabilities, for example,

- The Army Experimental Unmanned Ground Vehicle Systems (Demo III)
- Defense Advanced Research Projects Agency (DARPA)/Army Future Combat Systems
- DARPA Mobile Autonomous Robot Software
- DARPA Software for Distributed Robotics
- DARPA Tactical Mobile Robots
- National Aeronautics and Space Administration (NASA) spacecraft and rovers
- Department of Energy (DOE) waste remediation robot systems
- Department of Transportation (DOT) Intelligent Vehicle Initiative

In addition to the examples above, there are myriad other efforts in academia, industry, and government labs of work referred to as "intelligent systems." Despite the common use of "intelligent system" and "intelligent control," there is no uniform definition for either term. Generally, they are characterized by having one or more of the following traits [1]:

- Adaptive
- Capable of learning
- "Does the right thing" or "acts appropriately"
- Non-linear
- Autonomous symbol interpretation
- Goal-oriented
- Knowledge-based

These terms are ambiguous and qualitative. The Intelligent Systems Division of the National Institute of Standards and Technology has launched an initiative to better define what an intelligent system is and how to measure its performance. The mission of the Intelligent Systems Division, one of five divisions in the Manufacturing Engineering Laboratory, is "to develop the measurements and standards infrastructure needed for the application of intelligent systems by manufacturing industries and government agencies."

We are working with various industry groups and government agencies to tackle the issue of intelligent system performance. The Performance Metrics for Intelligent Systems Workshop is a foundational step, which brings together a multi-disciplinary community to help define the highest priority areas to concentrate on, having the highest payoff.

2. THE CHALLENGE OF DEFINING AND MEASURING MACHINE INTELLIGENCE

Researchers have been pursuing forms of machine intelligence for several decades. There have been many areas of focus, such as natural language understanding, expert systems to aid diagnoses, and decision-making tools for financial systems. Closer to our domain of interest, much effort has been focused on defining intelligent control as a discipline, but even so, there are no quantitative measures.

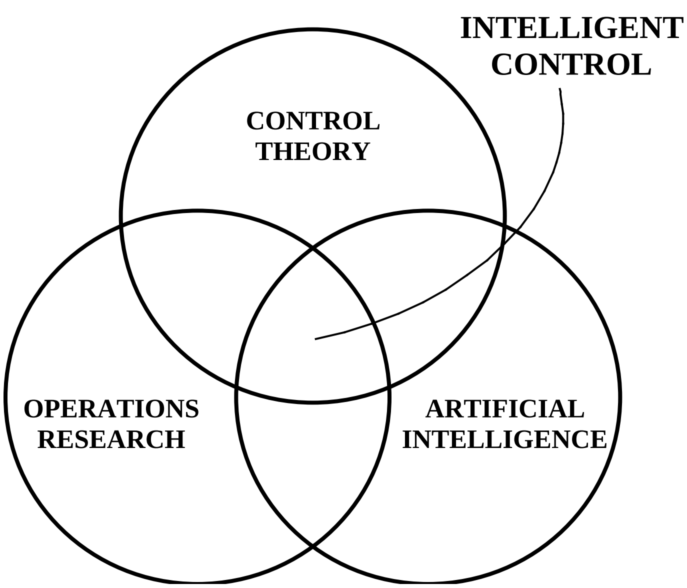


Figure 1: Intelligent Control as of 1985

Beginning with the efforts of Fu [1] and Saridis [3] in the seventies, there have been numerous conferences and workshops aimed at the topic of intelligent control. Nevertheless, the field remains fragmented due to its multidisciplinary nature. As noted in the first Symposium on Intelligent Control in 1985, intelligent control was proclaimed a theoretical domain, in which control theory, AI, and operations research intersected (Fig. 1 from [6]).

The definition of an intelligent system may be considered broader than that of intelligent control. As a "system," there may be more constituent parts, such as perception, world modeling, or value judgement. Yet more disciplines are brought into the picture. Examples of these include data representation, image processing, and decision theory.

Given the multi-disciplinary nature of the systems we are concerned with, it is clear that *defining the scope and performance of these systems is a challenge.* Terminology is one of the first hurdles that must be overcome. Different disciplines ascribe different definitions to the same words. For example, "complexity" may refer to non-linear systems in one field and to computational resources needed in another.

It is very difficult, if not impossible to currently evaluate research into intelligent systems. Since there are no quantitative metrics, intercomparisons of results are not generally possible. Sponsors are not able to adequately judge whether research results meet their requirements. Potential users have no impartial evaluation reports, *a la* "Consumer Reports," of intelligent systems, techniques, and tools. In general, the lack of metrics slows progress. There is a proliferation of data specific algorithms and task-specific solutions.

One of the biggest costs paid is the duplication of effort. New programs may be unable to have a firm definition of past accomplishments, hence they may fund work that repeats previous research. Research teams cannot leverage prior existing work from other institutions and tend to have to reinvent the wheel by building all of their system's components from scratch. They are burdened with having to spend effort in building components that are not part of their research focus, instead of being able to leverage existing "best of class" solutions and focussing on their interests.

Another negative impact, from the sponsor's viewpoint, is the lack of predictive ability in assessing new applications. Without objective performance evaluation metrics and an understanding of capabilities and limitations, it is difficult or impossible to assess claims of competing approaches in formulating new projects and programs. This leads to inefficiencies and failures that could be avoided if we had the measurement tools that we need.

3. ISSUES IN MEASURING PERFORMANCE

Numerous questions must be answered when considering how to define the performance of these intelligent systems. We will present a few questions. Many more will arise as we delve into the matter more closely.

- Should we measure only the external behavior of a system? Is that the only aspect that can feasibly be measured? Or, is there value in decomposing a system into components and measuring their individual capabilities? Examples would be measuring the path planning algorithms in isolation from the perception and other control subsystems.

- How generic does the measure of a system's intelligence have to be? Should we strive for general intelligence metrics that are domain-independent or are we better off focussing on application and domain-specific metrics? Are domain-independent metrics even meaningful?

- How do we factor in "body intelligence," the mechanical capabilities of a system as opposed to the control capabilities, when assessing the performance of a system? If we have a mobile robot, some of its abilities to achieve its stated goal (e.g., traverse a rubble pile to find survivors) can be attributed to its mechanical properties rather than its software intelligence.

- Are testbeds a viable measure of performance, or do they invite "gaming," that is, encourage solutions that are tailored to performing well in the testbed? If we don't have testbeds, how can we achieve reproducible measures of performance?

4. INITIAL OBSERVATIONS

One of the complicating factors in discussing intelligent systems is the use of the word "intelligence." It is freighted with significance and analogies to human or biological intelligence naturally arise. The quest for standard, uniform measures of intelligence in biological systems remains a subject of controversy. Therefore, we would advocate avoiding the temptation to spend too much time striving for performance measures that are based on human or higher level biological systems.

Observing that we are dealing with multi-disciplinary technologies and multiple application domains, we should expect that no single, unique measure of performance is feasible. Therefore, no single overarching and generic intelligence test will suffice. We need to strive for the right granularity of metrics.

We must be prepared to attack the problem on multiple fronts. It probably won't suffice to have just a theoretical investigation or an experimental one. Research must proceed on the theory as well as on gathering experimental data.

One of the key attributes of intelligent systems is its multi-disciplinarity. This poses a challenge, but also an opportunity. We can come together from a variety of disciplines and form a new community in which we share our expertise. We must have dialog and information exchange amongst ourselves in order to synthesize the best results from the different fields that contribute towards intelligent systems research.

That is the purpose of this workshop and the reason for the diversity of the presentations that you will hear.

5. CALL TO ACTION

The challenge is thus to define performance measures for new and evolving intelligent systems technologies that can greatly improve industrial productivity and advance government mission objectives. We must work together to build a technical foundation for measuring performance. This includes agreeing on the domains to investigate and a common set of terminology. We must develop theoretical foundations, methodologies, and supporting infrastructure for achieving our goals. Ultimately, measures must be developed that are practical, unambiguous, easy to use and widely deployable. We must simultaneously focus on attainable goals and strategies for both near-term and long-term measures of performance, as our understanding of them and the capabilities of the systems themselves evolve. Researchers, industry, and government will benefit from practical solutions they can readily apply, not from philosophical ones.

6. REFERENCES

[1] P. J. Antsaklis, "Defining Intelligent Control", Report of the Task Force on Intelligent Control, P.J Antsaklis, Chair, IEEE Control Systems Magazine, pp. 4-5 & 58-66, June 1994.

[2] K.-S. Fu, "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control," IEEE Trans. on Automatic Control, Vol. AC-16, 1971.

[3] Integrated Manufacturing Technology Roadmapping Initiative, “”Manufacturing Processes and Equipment Draft Roadmap,” Oak Ridge, TN, December, 1998.

[4] Neal, R. and Messina, E., ed., Proceedings of the First Part Correct Workshop, Gaithersburg, MD, April 2000.

[5] G. Saridis, Self-organizing Control of Stochastic Systems, M. Dekker, NY, 1977.

[6] G. N. Saridis, "Foundations of the Theory of Intelligent Controls," Proc. of the IEEE Workshop on Intelligent Control, Eds. G. Saridis, A. Meystel, Troy, NY 1985, pp. 23-28.

July 18, 2000

The Search for Metrics of Intelligence -- A Critical View

Lotfi A. Zadeh*

Few issues in AI generate as much heated debate as those which in one way or another relate to the questions: "What is intelligence?"; "Can machine think?"; and "How can intelligence be measured?" One cannot but be greatly impressed by the incisive comments made by members of the Intelligence Advisory Board. And yet, most of the basic issues relating to intelligence remain unresolved -- as they were half a century ago -- when I moderated, at Columbia University, what I believe to have been the first debate on "Can machines think?" The debate involved Claude Shannon, E.C. Berkeley, the author of Giant Brains, and Professor Francis J. Murray -- a prominent mathematician who as a consultant to IBM was active in the conception and design of computer systems.

At that time -- the dawn of the computer age -- there was a great deal of interest in the ability or inability of computers to think as humans do. To a much greater degree than is the case now, there were exaggerated expectations. In an article of mine entitled "Thinking machines -- a new field in electrical engineering," which appeared in the January, 1950, issue of the Columbia Engineering Quarterly (Zadeh 1950), I surveyed some of the articles which were published in the popular press at that time. The headline of one of the articles read "Electric brain capable of translating foreign languages is being built." The problem of machine translation seemed to be

* Professor in the Graduate School and Director, Berkeley Initiative in Soft Computing (BISC), Computer Science Division and the Electronics Research Laboratory, Department of EECS, University of California, Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712; E-Mail: zadeh@cs.berkeley.edu; Webpage: http://www.cs.berkeley.edu/People/Faculty/Homepages/zadeh.html.
Research supported in part by ONR Grant FDN0014991035, NASA Grant NAC2-1177, ONR Grant N00014-96-1-0556, ARO Grant DAAH 04-961-0341 and the BISC Program of UC Berkeley.

close to solution. Today, we know better. In 1997, Martin Kay, one of the leading contributors to machine translation had this to say: "Machine translation gave the initial inspiration to computational linguists and continues to motivate much of their work. That is surely fair enough since the problem is clearly computational and obviously linguistic. But forty years of money and effort has brought us hardly any closer to the answer. The world continues to pour money down the same rathole with little discernible progress, with or without the linguists. The German government is giving it a new twist: "Notice how we never seem to get anywhere on machine translation?"

The debates which raged in the past were largely of academic interest because there were few, if any, systems that could be assessed as having a high level of intelligence. At this juncture, this is no longer the case. Today, we can point with pride to Deep Blue, which beat Gary Kasparov. More importantly, we have a wide variety of systems which can perform highly non-trivial tasks involving recognition, decision and control. We are, in fact, witnessing the beginning of what may be described without exaggeration as the Intelligent Systems Revolution.

When AI was christened in 1956, it became the standard bearer of efforts to devise and build machines that could exhibit human-like intelligence in performing various tasks. For some time thereafter, the AI scene was one of unbridled enthusiasm and, as we now realize, unrealistic expectations. In judging that period, however, what should be remembered is that -- as Jules Verne astutely observed at the turn of the century -- scientific progress is driven by exaggerated expectations.

It took forty years for a computer to challenge and beat a chess champion. Why did it take so long to achieve some of AI's objectives? In the first place, the basic difficulty of approximating to what humans can do so easily without any measurements and any computations, e.g., understand speech, read handwriting, summarize a story and park a car, was greatly underestimated. More important, however, is the fact that the needed technologies and methodologies were not in place. In particular, we did not have the highly capable sensors and powerful computers which we have today, and we did not employ such recently developed methodologies as neurocomputing, evolutionary computing, probabilistic computing, machine learning and fuzzy logic.

In the past, what were called intelligent systems were for the most part symbol-manipulation oriented, e.g., machine translation systems, text understanding systems and game playing systems, among others. Today, what we see is the rapidly growing visibility of systems which are sensor-based and have embedded intelligence, e.g., smart washing machines, smart air conditioners, smart rice cookers and smart automobile transmissions. The counterpart of the concept of IQ in such systems is what might be called Machine IQ, or simply MIQ (Zadeh 1994). However, what is important to recognize is that MIQ -- as a metric of machine intelligence -- is product-specific and does not involve the same dimensions as human IQ. Furthermore, MIQ is relative. Thus, the MIQ of, say, a camera made in 1990 would be a measure of its intelligence relative to cameras made during the same period, and would be much lower than the MIQ of cameras made today.

Viewed in this perspective, the focus of activity in applications of machine intelligence is shifting from writing computer programs that can prove difficult theorems, understand text, provide expert advice and beat a chess champion, to more mundane tasks devolving on the conception, design and construction of products and systems that have a high MIQ, making them reliable, capable, affordable and user-friendly. Among recent examples of systems of this kind are programs which can detect the presence of known or new viruses in computer programs; checkout scanners which can identify fruit and vegetables through the use of scent sensors; car navigation systems which can guide a driver to a desired destination; password authentication systems employing biometric typing information; ATM eyeprint machines for identity verification; and molecular breath analyzers which are capable of diagnosing lung cancer, stomach ulcers and other diseases.

If MIQ is accepted as a metric of machine intelligence, then a particular machine may be said to be highly intelligent if has a high MIQ. But this beg the question of how the MIQ of a class of machines could be measured. Comments made by members of the Intelligent Advisory Board provide some guidelines. But a thesis that I should like to put on the table is that the existing conceptual framework of AI -- which is based on first-order two-valued logic -- is incapable of providing a suitable foundation for constructing realistic metrics of IQ and MIQ.

The problem with predicate-logic-based AI is that it embraces the principle of the excluded middle, which asserts that every proposition is either true or false, with no shades of gray allowed. But in the real world, as perceived by humans, it is partiality rather than categoricity that is the norm. Thus, we generally deal with partial knowledge, partial order, partial truth, partial certainty, partial causality and partial understanding. The essentiality of the role of partiality in human cognition has been slow in gaining acceptance in AI. Without employing the notion of partiality, realistic metrics of IQ and MIQ cannot be constructed.

Another concept that plays a basic role in human cognition is that of granularity, and, more particularly, that of f-granularity. In essence, f-granularity is a concomitant of the bounded ability of sensory organs and, ultimately, the brain, to resolve detail and store information. What this means is that (a) the boundaries of perceived classes are not sharply defined; and (b) values of perceived attributes are granulated, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. For example, the granules of Age might be: very young, young, middle-aged, old and very old. Similarly, the granules of face may be: nose, cheeks, chin, forehead, etc. F-granularity underlies the concept of a linguistic variable in fuzzy logic.

The concepts of partiality and f-granularity play key roles in what may be called Precisiated Natural Language (PNL). What I should like to suggest is that PNL could play a central role in formulation of metrics of intelligence. How these could be done is a complex task that will require a major effort to yield concrete results. In what follows, I will confine myself to sketching the basics of PNL and pointing to its use as a concept definition language.

Natural languages are expressive but imprecise. Mathematical languages are inexpressive but precise. Basically, PNL draws on a natural language (NL) and a mathematical language (ML) to provide a language which is precise and yet far more expressive than conventional meaning-representation and definition languages based on predicate logic.

In essence, PNL is a subset of NL which consists of propositions which are precisiable through translation into a precisiation language GCL (Generalized Constraint Language). An example of a precisiable proposition is: It is very unlikely that there will be a significant increase in the price of oil in the near future. The point of departure in PNL is the assumption that the meaning of a precisiable proposition, p, is expressible as a generalized constraint on a variable. Usually, the constrained variable and the constraining relation are implicit rather than explicit in p.

A concept which has a position of centrality in GCL is that of a generalized constraint expressed as X isr R, where X is the constrained variable, R is the constraining relation, and isr (pronounced as ezar) is a variable copula in which r is a discrete-valued indexing variable whose value defines the way in which R constrains X. Among the principal types of constraints are the following: possibilistic constraint, r=blank, with R playing the role of the possibility distribution of X; veristic constraint, r=v, in which case R is the verity (truth) distribution of X; probabilistic constraint, r=p, in which case X is a random variable and R is its probability distribution; r=rs, in which case X is a fuzzy-set-valued random variable (fuzzy random set) and R is its fuzzy-set-valued probability distribution; and fuzzy-graph constraint, r=fg, in which case X is a fuzzy-set-valued variable and R is its fuzzy-set-valued possibility distribution.

With these constraints serving as basic building blocks, which are analogous to terminal symbols in a formal language, more complex (composite) constraints may be constructed through the use of a grammar. Simple examples of composite constraints are: X isr R and X iss S; and, if X isr R then Y iss S, or, equivalently, Y iss S if X isr R. The collection of composite constraints forms the Generalized Constraint Language (GCL). The semantics of GCL is defined by the rules that govern combination and propagation of generalized constraints. These rules coincide with the rules of inference in fuzzy logic (FL).

The capability of PNL to serve as a powerful definition language depends in large measure on the fact that, by construction, GCL is maximally expressive. The conclusion that emerges from this fact is that metrics of intelligence, if they can be defined, will necessarily have to be defined in terms of PNL and have an algorithmic structure (Zadeh 1976). What this implies is that realistic metrization of intelligence is not possible within the conceptual structure of existing methods of definition and measurement. We cannot expect a concept as complex as that of intelligence to be definable in traditional terms.

References

1. L.A. Zadeh, "Thinking machines - a new field in electrical engineering," *Columbia Engineering Quarterly 3*, 12-13, 30, 31, 1950.

2. L.A. Zadeh, "A fuzzy-algorithmic approach to the definition of complex or imprecise concepts," *Int. Jour. Man-Machine Studies 8*, 249-291, 1976.

3. L.A. Zadeh, "Fuzzy logic, neural networks and soft computing," *Communications of the ACM 37(3)*, 77-84, 1994.

Measure of System Intelligence: An Engineering Perspective

Sukhan Lee[†], Won-Chul Bang[‡] and Z. Zenn Bien[‡]

† System and Control Sector, SAIT, Korea

‡ Div. of EECS, Dept. of EE, KAIST, Korea

ABSTRACT

System intelligence can be measured experimentally either through benchmark tests, or theoretically through the formal analysis of system software architecture and hardware configurations. The latter approach is pursued here, since it serves directly as the criteria for designing and engineering intelligent systems in a directed manner, rather than by trial and error. To this end, a structure of problem solving and learning of machine is proposed. Once a machine is represented with the structure, the intelligence can be measured via transforming it into an equivalent linguistic structure. A simple example is also provided.

KEYWORDS: *measure of system intelligence, measure by linguistic equivalence, machine description language*

1. INTRODUCTION

The intelligence of systems is emergent when the systems are able to accomplish loosely defined but complex tasks in an unstructured and uncertain environment. The intelligence can be manifested by the capability of systems to autonomously synthesize goal-oriented behaviors in adaption errors, faults, and unexpected events through the real-time connection of sensing and action. However, we still do not have a satisfactory quantitative way to characterize the "intelligence" of systems. There are many kinds of intelligent systems in various fields. The adjective 'intelligent' is quite widely used to describe their systems developed by many system engineers and companies. One developer may say that his/her system is more intelligent than the others, but it can happen that another claims the same thing. In this case, who can say one is more intelligent than the others? One must have a kind of measure of intelligence for systems or machines in order to answer this question. In this sense, it is worthwhile to provide a measure on how intelligent a machine is.

Many intelligent system techniques have been developed and studied so far, but only a few studies have been done on '*how to measure intelligence of systems*.' J. S. Albus introduced the theory of intelligence in an engineering viewpoint [1]. G. Zames initiated an effort for defining such an index as approximate a measure of the "task" and "satisfactory" performances an "intelligent controller" could achieve versus those that a classical controller could achieve [2]. The challenge involves characterization of performance in unknown environments, learning, controller and task complexity, and associated tradeoffs. E. C. Chalfant and S. Lee suggested an engineering perspective [3]. They thought that one can represent all tasks of a machine in the form of graphs and find an equivalent language for the graphs. Since a language consists of grammar and vocabulary, the descriptive power of a machine can be represented by the grammar and the vocabulary. Bien, et al. [4][5] proposed a couple of methods to measure how much a machine is intelligent; they considered the questions from the ontological (functional) and phenomenological (behavioral) definitions on intelligent machine.

Establishing the measure of system intelligence should not only be able to turn the intelligent system into a formal academic discipline but also provide a means of designing better and more powerful intelligent systems in practice. The measure of intelligence of a system or, more precisely, a constructed system with autonomy should take into consideration various aspects of intelligence ranging from perception, understanding, and problem solving to generalization and learning from experience. A. Meystel proposed a vector of system intelligence as a collection of features representing intelligent functions of a system. The list of such features can be very comprehensive indeed. However, formulating the measure of system intelligence based on such a vector may not necessarily represent the essence of system intelligence. The functional features describing the aspect of intelligent behaviors may obscure the existing internal engine by which intelligent behaviors are generated.

To begin with, the following questions are raised for answer prior to the definition of the metric of system intelligence:

(a) Should the intelligence measure be goal-dependent or goal-independent?

(b) Should the intelligence measure be time-varying or time-invariant?

(c) Should the intelligence measure be resource-dependent or resource-independent?

For (a), it raises a question whether there exists a universal measure of system intelligence such that the intelligence of systems can be compared independently of the given goals. A goal-independent measure may be more difficult to define, if not impossible, and more controversial. A goal-dependent measure, however abstract the goal may be, can allow clear comparison among the systems of different architecture but with the same goal. For instance, for the latter case, intelligence can be represented as how efficiently, and how optimally a system reaches the given goal by itself, i.e., the power of automatically solving problems defined as the discrepancy between the goal and the current state.

For (b), it represents whether the intelligence measure of a system should solely be based on problem-solving capability at time t or it should contain the potential increase of problem-solving capability in the future based on learning. Both are necessary. But, it is better to define the two separately before integrating them together in one measure.

For (c), it raises an issue whether the resources required for building systems and system operation should play a role for defining the measure of intelligence. As mentioned above, the efficiency in problem solving should be included in the measure: for instance, the time and energy required to reach a solution should be taken into consideration together with the optimality of the solution. But, it is not clear whether we should or should not include the cost of building a system.

Section 2 provides definitions of engineering metric of system intelligence based on the above three questions. In Section 3, machine intelligence structure is proposed, and an equivalent linguistic structure follows in Section 4. Section 5 shows an example with a robotic arm. Finally, Section 6 concludes the paper.

2. DEFINITION OF ENGINEERING METRIC OF SYSTEM INTELLIGENCE

System intelligence can be measured under considering various points of views described in the previous section. An approach in engineering perspective is pursued here with *goal-oriented*, *time-dependent*, and *resource-dependent* definition of engineering metric of system intelligence. We define machine intelligence quotient (MIQ) in the following way.

The measure of system intelligence as problem-solving capability at time t for the given goal set g, denoted by $MIQ(g, t)$, is defined by the capability of solving problems toward the given goal set where the capability can be measured by the scope of constraints (environmental variations), together with the time and resources required, under which the system succeeds in reaching the given goals.

The measure of self-improvement of system intelligence as learning capability with respect to time t, denoted by $dMIQ(g, t)$, can be defined by the rate of increasing $MIQ(g, t)$ with respect to time based on learning from experience. Capability of learning in the time duration of (t_1, t_2) is represented by the integration of $dMIQ(g, t)$ between t_1 and t_2.

Now, the total measure of system intelligence, $tMIQ$, is defined by

$$tMIQ = \max_t [MIQ(g,t_0) + \int_{t_0}^{t} dMIQ(g,t)dt]. \qquad (1)$$

Let $tmax$ be the time when the maximum of $tMIQ$ is obtained. The learning rate is then defined by $\max_t \int_{t_0}^{t} dMIQ(g,t)dt \Big/ tmax$.

Note that the universal measure of system intelligence, $uMIQ$, may be defined in terms of integration of MIQ with respect to goal, i.e.,

$$uMIQ = \int_{g \in G} \max_t [MIQ(g,t_0) + \int_{t_0}^{t} dMIQ(g,t)dt] dg \qquad (2)$$

where G is the set of all goals.

As mentioned above, resources required for the machine is combined into the machine intelligence, MIQ to resource ratio, $rMIQ$, can be represented by

$$rMIQ = tMIQ / resources. \qquad (3)$$

3. MACHINE INTELLIGENCE

As described in the previous section, machine intelligence can be measured once $MIQ(g, t)$ and $dMIQ(g, t)$ are defined. We now formulate the way of defining two quantities, MIQ (problem-solving capability) and $dMIQ$ (rate of increasing MIQ based on learning capability).

The first step of problem solving is to understand the situation and define what are the problems to solve. This requires identifying the gap between the goal and current states as well as recognizing the constraints and opportunities imposed by the environment. Then follows the planning or decision-making to reduce the gap under constraints. The first step requires perception and understanding, whereas the second step requires action and planning. Perception and action can be represented as logical sensor and actuator systems, respectively, in a form of hierarchical graphs of declarative knowledge components. Understanding can be represented as the connection of what have been perceived to system internal knowledge. Planning can be represented as the projection of what have been understood to the logical actuator system. The mechanism of these connections can be rule-based. The overall structure of problem solving mechanism is represented in Figure 1 with solid-line connections.

Regarding the learning capability, a higher level of consciousness that monitors these activities of understanding and planning may exist in the form of thinking (a self-driven function that monitors understanding and planning in the form of questioning, virtual manipulation). In case that the machine cannot understand an obtained data from logical sensors by perception, the consciousness/emotion may adjust the knowledge to allow the obtained data for understanding, i.e., identifying the gap between the goal and current states as well as recognizing the constraints and opportunities imposed by the environment. In addition, when an action already taken is decided to be further improved, the consciousness/emotion may fix its knowledge to give a better plan later on. The structure of learning mechanism is also shown in Figure 1 with dotted-line connection.

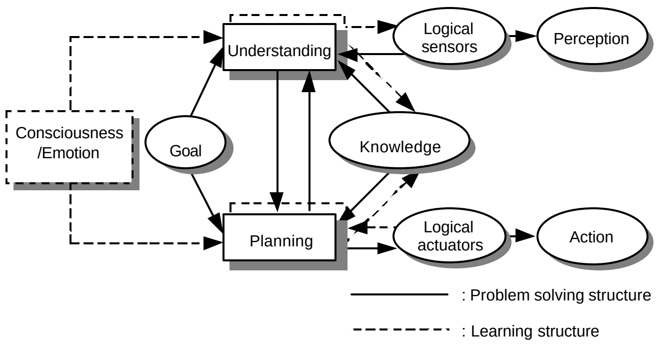


Figure 1. Structure of Machine Intelligence

The logical sensors and actuators as well as knowledge and constraint can be represented by an equivalent linguistic form. The same is true for representing the connection and projection associated with understanding and planning. If the functions of a system embedded in its hardware and software can be represented as a linguistic equivalent, based on the above observation, the MIQ and dMIQ of the system may be defined in the equivalent linguistic space. Thus, for a given machine to measure its intelligence, transforming the machine itself into this structure of problem solving and learning is first conducted, and then transforming it into the equivalent linguistic structure is to be done, which is discussed in the next section.

4. MEASURE BY LINGUISTIC EQUIVALENCE

Transforming system architecture into an equivalent formal language structure, a consistent measure of machine intelligence associated with the corresponding formal language can be obtained.

Any generic language used to build models representing diverse architectures must contain mechanisms to implement the features of all these architectures. For example, the parallel structure of the subsumption model requires parallelism in the language. At the other extreme, the functionality of a centralized planner must also be representable. If the structure of the model differs, we must be prepared to clearly determine equivalent operation.

4.1 The Machine Description Language

The basic unit of the Machine Description Language (MDL) is a behavior. The behavior nit is analogous to a sentence or statement constructed according to grammatical rules. There statements are conglomerated to form a meaningful system. The paper defines the grammatical rules of syntax of the Machine Description Language. Generating the semantics of an entire system is analogous to writing a program in a given system.

An MDL model has a hierarchical layered architecture composed of a number of various behaviors, some simple, and some complex. The simplest possible behavior is based on direct triggering by a single binary sensor which elicits a simple actuator response. For example, an on/off contact switch can trigger a behavior called “bump” which causes a short reverse movement combined with a turn.

Behavior modules are collected in groups which implement a complete autonomous task, such as obstacle detection. The collection of behaviors is called a *wrapped behavior.* The linguistic analogy is a paragraph of subroutine which encapsulates a single topic or function.

The composite wrapped behavior collectively implements some useful autonomous task. For example, a group of bump behaviors based on different contact sensors can be wrapped to form an obstacle rerouting wrapped behavior based on direct contact. If ultrasonic range detectors are added, new strands can be added to the composite object rerouting behavior, and the improved behavior them before bumping them. The old bump behaviors are kept as backups.

4.2 Analytical Measures with MDL

The performance of the system described here can be measured using traditional back box empirical techniques. For example, we can time its performance in executing a prescribed task. Alternatively, structural (linguistic) analyses of the system can be used to determine theoretical bounds on performance independent of implementational efficiency.

Structural analysis begins with identification of measurable quantities and their effects on performance. Many structural features can be measured; each contributes to the emergent intelligence of the completed system in a different way.

4.2.1 Behavior Attributes

We first consider measurable attributes of a behavior. Some of the measurable structural features are:

Strand Count and Strand Segment Count: A behavior has some number of strands (i.e., sensor to actuator information path) associated with it. Strands are regarded as instantaneous communication links for the purpose of measurement. The information packet propagation time between nodes, trigger, and taps is zero. The number and thickness of strands in a single behavior provides a measure of the resolution of sensory information, trigger situation discernability, and the dexterity or controllability of the actuator system. More fundamentally, strand *segment* count and thickness together measure the information transport capacity of the behavior.

Node Count: Node count captures the complexity of the sensor and actuator trees of a behavior. The node count is taken as the sum of nodes and taps for both sensor and actuator trees.

Trigger Propagation Time: Each trigger has three measurable attributes indicating the dimensionality of the input (parameters of the sensed situation), the dimensionality of the output (parameters of the desired response, based on the sensed situation) and the propagation time of the information, i.e., the delay between a sensed situation and the resultant response.

Node Propagation Time: The delay an information packet encounters between the time it enters a tap node, fusion node, or arbitration node and the time it (or the effects of a change in the information) exits the node, is termed *node propagation time*. It represents the processing time required to fuse information, to arbitrate competing controls, or to extract or combine information.

Strand Propagation time: The strand propagation time id the time for an information packet to travel from the sensor at the beginning of the strand to the actuator at the end of the strand.

Behavior Response Time: The response time of a behavior is the sum of all information propagation timers along the longest path between raw sensor input and raw actuator output. The path may include nodes from other behaviors but will include only one trigger propagation time. This differ from the propagation time of the longest strand in that the strand propagation time is measured from tap to tap, whereas the behavior response time is measured from raw sensor input to raw actuator output. Behavior response time is computed as:

$$\boldsymbol{B} = \max_i (\sum a_i + t) \qquad (4)$$

where

$\boldsymbol{B}$: behavior response time

a_i : node propagation time for node i

t : trigger propagation time

Behavior response time can also be measured empirically, as long as the response can be isolated from the response of all other behaviors.

4.2.2 System Attributes

Next we consider attributes of the combined system:

Trigger of Behavior Count: The number of separate triggers (which is equivalent to the number of behavior modules) indicates the number of separate situations and corresponding responses, which the system can elicit, based on its sensory information. The total number of triggers in the entire system is and indication of complexity of the system and sophistication of response (assuming a well-designed system).

Strand Distribution: Strands which rely on many lower level strands provide more abstract, goal-directed, and strategic stimulus-response relationships, whereas the lower level strands provide greater reactivity and quicker response. The distribution of the strands between these extremes indicates the tendency for the system to generate behavior based on reflexes or impulses vs. goal-seeking behavior. One measure of this characteristic is the distribution of behavior propagation times. Standard statistical measures such as mean and median behavior propagation times, standard deviation, minimum and maximum propagation, describe the distribution. A median propagation time biased toward the minimum indicates a more quickly responsive and reflexive system whereas a bias toward the maximum indicated a deliberative system.

Layering Depth: Another measure of deliberativeness is the layering depth. The layering depth can be measured as the number of trees belonging to different behaviors which an information packet must traverse to reach the raw motors from the trigger. Because each group of wrapped behaviors comprises an autonomous set of behaviors, the layering depth or maximum depth of wrappers indicated the sophistication of autonomy. A system, which is more deeply wrapped, may indicate that it can perform more complex tasks autonomously. Each behavior added to a wrapped behavior indicates that some environmental situation can arise which s not handled optimally by the wrapped behavior by itself. If a wrapped behavior s itself wrapped along with new behaviors, the newly wrapped set handles all the environmental stimuli of the original wrapper plus all the situations detected by the new behviors.

MIQ: The *MIQ* (Machine Intelligence Quotient) is then defined as the product of the complexity of tasks the system can handle and the performance in task execution. This measure embodies the tradeoff between reflexivity (speed) and deliberativity (complexity). Task complexity is dependent both on the complexity and quantity of the tree structures. The complexity of tasks can be measured using the system attributes listed above, namely, trigger count, strand distribution, layering depth, strand count, and node count. We combine these as a weighted sum:

$$\boldsymbol{T} = w_g g + w_d d + w_l l + w_s s + w_k k \qquad (5)$$

where

$\boldsymbol{T}$: Task complexity ability

g : Trigger count

d : Average strand propagation time overall machine

l : Layering depth

s : Total strand count in machine

k : Total node count in machine

w_g, w_d, w_l, w_s, w_k : Respective weights

Performance in task execution is derived from the collective performance of behaviors. This can be computed as the weighted sum of behavior response time and *inverse* average strand propagation time (since speed increase as strand length decreases):

$$\boldsymbol{E} = w_B \boldsymbol{B} + w_d / d \tag{6}$$

MIQ is then

$$MIQ = \boldsymbol{T} \cdot \boldsymbol{E} \tag{7}$$

Resource: Machine "resource" is a measure of implementation requirements based on the architectural design of the machine. The resource is defined as the product of cost and volume. We compute the resource based on the number of processors and communication links required to implement the system directly in a parallel architecture. Processors are expensive while communication links are cheap. However, communication links can become numerous and occupy a large part of the volume of a machine. These costs and volumes are likely to change with new technology. The cost of the system is the sum of the costs of the processors (trigger, nodes, and taps) required. We assume one simple processor per trigger, node, or tap. We denote this as

$$\boldsymbol{C} = C_g g + C_p p \tag{8}$$

where

$\boldsymbol{C}$: cost of machine

p : node count

C_g, C_p : cost of trigger and node processors

The volume of the system is computed the same way:

$$\boldsymbol{V} = V_g g + V_p p \tag{9}$$

Then resource is

$$\boldsymbol{R} = CV \tag{10}$$

and the *rMIQ* is

$$rMIQ = MIQ / \boldsymbol{R} \tag{11}$$

5. ENGINEERING CASE STUDY

A simple grasp controller based on the subsumption style of robot control uses a gripper beam and finger contacts as sensors as shown in Figure 2.

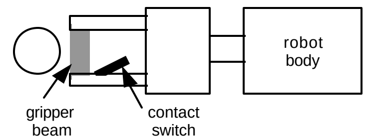


Figure 2. A Simple Robot Arm

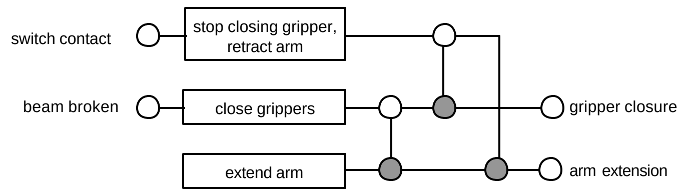


Figure 3. Subsumption Network

Figure 3 illustrates the simple subsumption network which generates the behavior of the robot. The extend arm behavior is always extends the arm (we ignore the condition of a fully extended arm). As soon as the gripper beam is broken, the sensor causes the "close grippers" behavior to trigger. The white motor node simultaneously inhibits the arm from extending with an inhibition node and activates the gripper closure actuator, causing the gripper to begin closing. (The gray nodes are taps – in this example they are motor taps or arbitrators.) When the grippers contact the object, the contact switch is closed, causing the "stop closing gripper, retract arm" behavior to trigger. The white node on the output of this behavior is a sequential node – first the gripper closure motor strand is inhibited, causing the gripper to first stop squeezing. Finally, the behavior subsumes the output of the "extend arm" behavior using a subsumption node, causing the arm to retract.

The *MIQ* and *dMIQ* of this system is easy to compute. All weights are set to one to simplify the example. There are three behaviors. The "extend arm" behavior is a trigger and a raw motor node (the tap nodes belong to the /"close gripper" and "stop gripper…" behaviors). The behavior response time for "extend arm" is therefore 1 + 1 = 2. There is one strand in this behavior. The "close grippers" behavior has one raw sensor node, one motor node tree node, and either one raw motor node or one motor tap; both of the two strands are the same length, so we may use either. The response time is 3 + 1 = 4. The "stop closing…" behavior similarly has a response time of 4 and a strand count of two. The mean behavior response or propagation time is (2 + 4 + 4) / 3, or 3.333. Layering depth is two, and system strand count is 5. Average strand propagation time over the entire system is (3 + 3 +3 + 3 + 1) / 5, or 2.6. There are nine nodes and nine strand segments in the entire system.

Based on these numbers, task complexity ability is 3 + 2.6 + 2 + 5 + 9 = 21.6. Remember, this number means little except as a comparative measure. Performance is 3.333 + 0.385 = 3.718. MIQ is then roughly 21.6 + 3.7 = 25.3. If we assume costs and volume of one, then cost and volume are both 9 + 9 = 18. Resource is (18)(18) = 324, and the rMIQ is 21.6/324 = 0.0667

6. CONCLUSION

We have presented three important issues, which should be considered when measuring machine intelligence, and introduced the structure of machine intelligence, which shows the internal mechanism of machine taking into account the three issues. Any machine can be represented by the proposed structure and the structure can be transformed into an equivalent linguistic structure so that one may define the metric of the machine intelligence in an analytical way.

In this paper, an equivalent linguistic structure has been proposed. It needs to be further developed to present linguistic structure of machine intelligence for both *MIQ* and *dMIQ* with respect to goals and time.

The formulation on *MIQ*, *dMIQ*, and *rMIQ* in Section 2 will be a good guide for defining machine intelligence since its clearness in the sense of goal-dependency, time-varyingness, and resource-dependency.

REFERENCES

[1] Albus, J. S., “Outline for a Theory of Intelligence,” *IEEE Tran. on Systems, Man, and Cybernetics*, Vol. 21, No. 3, 1991, pp. 473-509

[2] Antsaklis, P. J. “At the gates of the Millennium: Are we in control?,” *IEEE Control Systems Magazine*, Vol. 20, No. 1, February 2000, pp. 50-55

[3] Chalfant, E. C., and Lee, S., “Measuring the Intelligence of Robotic Systems: An Engineering Perspective,” *Proceedings of International Symposium on Intelligent Systems*, Gaithersburg, MD, October, 1999

[4] Bien, Z., Kim, Y.-T., and Yang, S., “How To Measure The Machine Intelligence Quotient (MIQ): Two Methods And Applications,” *Proceedings of World Automation Congress*, Alaska, May 9-14, 1998

[5] Bien, Z., Bang, W.-C., and Han, J.-S., “A Perspective of Intelligent Machine and Measurement of Machine Intelligence Quotient,” *Proceedings of the 4th Korea-Japan Joint Workshop on Advanced Teleautomation and Intelligent Mechatronics*, Taejon, Korea, Nov. 4-6, 1998, pp. 64-67

[6] Lee, S., Schenker P. S., and Park, J., “Sensor-Knowledge-Command Fusion Paradigm for Man/Machine Systems,” *Proceedings of SPIE International Symposium on Advanced Intelligent Systems*, Boston, MA, 1990

Formal Specification of Performance Metrics for Intelligent Systems

Ying Zhang
System and Practice Lab, Xerox Palo Alto Research Center
Palo Alto, CA 94304
Alan K. Mackworth
Department of Computer Science, University of British Columbia
Vancouver, B.C., Canada, V6T 1Z4
Email: yzhang@parc.xerox.com, mack@cs.ubc.ca

ABSTRACT

There are now so many architectures for intelligent systems: deliberative planning vs. reactive acting, behavioral subsuming vs. hierarchical structuring, machine learning vs. logic reasoning, and symbolic representation vs. procedural knowledge. The arguments from all schools are all based on how natural systems (i.e., biologically inspired, from basic forms of life to high level intelligence) work by taking the parts that support their architectures. In this paper, we take an engineering point of view, i.e., by using requirements specification and system verification as the measurement tool. Since most intelligent systems are real-time dynamic systems (all lives are), requirements specification should be able to represent timed properties. We have developed timed ∀-automata that fit to this purpose. We will present this formal specification, examples for specifying requirements and a general procedure for verification.

KEYWORDS: *formal specification, constraint-based requirements, system verification*

1. INTRODUCTION AND MOTIVATION

Over the last half a century, intelligent systems have become more and more important to human society, from everyday life to exploration adventures. However, unlike most other engineering fields, there has been little effort towards developing sound and deep foundations for quantitatively measurement and understanding such systems. The lack of measurement and understanding leads to unsatisfactory behavior or even potential danger for customers. The systems may not achieve desired performance in certain environments, or, the systems may even result in catastrophe in life-critical circumstances.

Many researchers have suggested measures of performance for intelligent systems, such as the Turing Test [12], Newell's expanded list [9,10] and Albus's definition of intelligence [4]. However, most of these measures are not based on formal quantitative metrics. There are also efforts on comparing performance on pre-defined tasks, such as a soccer competition [11]. However, these methods are domain specific therefore hard to apply to general cases. We advocate formal methods for specifying performance requirements of intelligent systems. Much research has been done on formal methods (http://archive.comlab.ox.ac.uk/formal-methods.html) over the last twenty years. In this paper, we explore one of the approaches, namely, using timed ∀-automata for specifying performance requirements.

The timed ∀-automata model was developed in [13, 17] as an extension of discrete time ∀-automata [8] to continuous time, annotations with real-time. Timed ∀-automata are simple yet able to represent many important features of dynamic systems such as safety, stability, reachability and real-time response. In the rest of this paper, we introduce the formal definition of timed ∀-automata first, then present examples of timed ∀-automata for representing performance metrics, and finally describe a general verification procedure for this type of requirements specification.

2. TIMED ∀-AUTOMATA

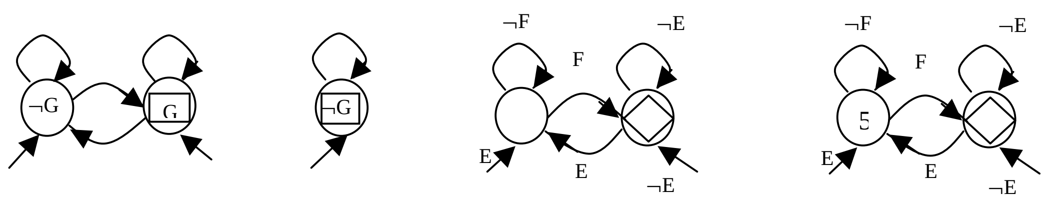
In general, there are two uses of automata: 1. to describe computations, such as input/output state automata, and 2. to characterize a set of sequences, such as regular grammars/languages. Examples of the first category are mostly deterministic and examples of the second category are mostly non-deterministic. However, all the original automata work is based on discrete time steps/sequences. Approaches to extending automata to continuous time have been explored in hybrid systems community over the last decades [1,2,7]. The timed ∀-automata model that we developed belongs to the second category, i.e., *non*-deterministic *finite* state automata specifying behaviors over *continuous* time. The discrete time version of ∀-automata was originally proposed as formalism for the specification and verification of temporal properties of concurrent programs [8].

2.1. *Syntax*

Syntactically, a timed ∀-automaton is defined as follows.

[**Definition 1**] A ∀-*automaton* A is a quintuple (Q, R, S, e, c) where Q is a finite set of automaton-states, $R \subseteq Q$ is a set of recurrent states and $S \subseteq Q$ is a set of stable states. With each $q \in Q$, we associate an assertion e(q), which characterizes the entry condition under which the automaton may start its activity in q. With each pair q, q' $\in$ Q, we associate an assertion c(q, q'), which characterizes the transition condition under which the automaton may move from q to q'. R and S are generalizations of accepting states. We denote by $B = Q - (R \cup S)$ the set of non-accepting (bad) states. Let R^+ be the set of non-negative real numbers representing time durations. A *timed ∀-automaton* is a triple (A, T, τ) where A is a ∀-automaton, $T \subseteq Q$ is a set of timed automaton-states and $\tau: T \cup \{B\} \rightarrow R^+ \cup \{\infty\}$ is a time function.

One of the engineering advantages of using automata as a specification language is its graphical representation. It is useful and illuminating to represent timed ∀-automata by diagrams. A timed ∀-automaton can be depicted by a labeled directed graph, where automaton-states are depicted by circle nodes and transition relations are by directional arcs. In addition, each automaton-state may have an entry arc pointing to it; each recurrent state is depicted by a diamond and each stable state is depicted by a square, inscribed within a circle. Nodes and arcs are labeled by assertions as follows. A node or an arc that is left unlabeled is considered to be labeled with **true**. Furthermore, (1) if an automaton-state q is labeled by ψ and its entry arc is labeled by φ, the entry condition e(q) is given by $e(q) = \psi \wedge \varphi$; if there is no entry arc, e(q) = **false**, and (2) if arcs from q to q' are labeled by φ_i, i = 1…n, and q' is labeled by ψ, the transition condition c(q, q') is given by $c(q, q') = (\varphi_1 \vee \ldots \vee \varphi_n) \wedge \psi$; if there is no arc from q to q', c(q, q') = **false**. A T-state is denoted by a nonnegative real number indicating its time bound. Some examples of timed ∀-automata are shown in Figure 1.



(a) (b) (c) (d)

Figure 1. Examples of timed ∀-automata

2.2. *Semantics*

Semantically, each assertion denotes a constraint defined on a domain of interest. Let D be a domain of interest; D can be finite, discrete, or continuous, or a cross product of a finite number of domains. Physically, D can represent, for example, speeds, distances, torques, sentences, commands or a combination of the above. A constraint C defined on D is a subset of D, $C \subseteq D$. Physically, a constraint represents certain relation on a domain, such as a relation between external environment stimuli and an agent's internal knowledge representation, or, a relation between internal states and actions, or, the relation between the current and next state. An element d in domain D satisfies constraint C, if and only if $d \in C$.

The semantics of timed ∀-automaton is defined as follows. Let *T* be a time domain, which can be continuous, for example, R^+. First, let us define runs of ∀-automata. Let A = (Q, R, S, e, c) be a ∀-automaton and $v: T \rightarrow D$ be a function of time. A *run* of A over v is a function $r: T \rightarrow Q$ satisfying:

1. *Initiality*: $v(0) \in e(r(0))$;
2. *Consecution*:
   a. *Inductivity*: $\forall t>0$, $\exists q \in Q$, $t'<t, \forall t''$, $t' \leq t'' < t$, $r(t'')=q$ and $v(t) \in c(r(t''), r(t))$ and
   b. *Continuity*: $\forall t$, $\exists q \in Q$, $t'>t$, $\forall t''$, $t<t''<t'$, $r(t'')=q$ and $v(t'') \in c(r(t), r(t''))$.

   When *T* is discrete, the two conditions in *Consecution* reduce to one, i.e., $\forall t>0$, $v(t) \in c(r(pre(t)), r(t))$ where pre(t) is the previous time point of t.

If r is a run, let Inf(r) be the set of automaton-states appearing infinitely many times in r, i.e., $Inf(r) = \{q | \forall t \exists t' \geq t, r(t')=q\}$. A run is called *accepting* if and only if

1. $Inf(r) \cap R \neq 0$, i.e., some of states appearing infinitely many times in r belong to R, or
2. $Inf(r) \subseteq S$, i.e., all the states appearing infinitely many times in r belong to S.

For a timed ∀-automaton, in addition for a run to be accepting, it has to satisfy time constraints. Let $I \subseteq T$ be a time interval and |I| be the time measurement, and let r|I be a segment of r over time interval I. A run satisfies time constraints if and only if:

1. *Local*: For any $q \in T$ any time interval I, if r|I is a segment of consecutive states of q, then $|I| \leq \tau(q)$;
2. *Global*: For any time interval I, if r|I is a segment of consecutive states of $B \cup S$, then $\int_I \chi_B(r(t))dt \leq \tau(B)$, where $\chi_{B:}\ Q \rightarrow \{0,1\}$ is the characterization function for the set B.

[**Definition 2**] A timed ∀-automaton TA = (A, T, τ) accepts a trace v, if and only if

1. All runs are accepting for A;
2. All runs satisfy the time constraints.

With the semantics defined, we can infer that, for the timed ∀-automata in Figure 1, (a) specifies the behavior of reachability, i.e., eventually the system should satisfy constraint G, (b) specifies the behavior of safety, i.e. constraint G is never satisfied, (c) specifies the behavior of bounded response, i.e., whenever constraint E is satisfied, constraint F will be satisfied within bounded time and (d) specifies the behavior of real-time response, i.e., whenever constraint E is satisfied, constraint F will be satisfied within 5 time units.

3. EXAMPLES OF PERFORMANCE SPECIFICATION

Timed ∀-automata are simple yet powerful for the specification of behaviors of dynamic systems, since it integrates constraint specification with timed dynamic behavior specification.

3.1. *Examples of Constraint Specification*

Constraint specification alone can specify many performance metrics. Constraints specify relations between external environment stimuli and an agent's internal knowledge representation, or between internal states and actions, or between the current and next states. Constraints can be finite, discrete or continuous, or any combination of the above. Constraints can be linear, nonlinear, equalities or inequalities. Moreover, constraints can also specify optimal conditions or optimality with extra constraints, or combinations of multiple optimal criteria and additional constraints.

Considering the following examples for specifying constraints:

1. *Inequality:* $f(x) \leq 0$ where x is a vector of variables and f is a vector of functions.
2. *Optimality*: min $|f(x)|$ where $|x|$ is a norm for x.
3. *Negation*: $x \neq y$.
4. *Constrained Optimality*: $\min|f(x)|$ given $g(x) \leq 0$.
5. *Robustness*: Let f(x) be a set of output functions with x as inputs. The robustness can be represented by its Jacobian $J = \Delta f/\Delta x$. There are many ways to state an optimal condition for robustness. One method is to minimize $|w|$ where w is the diagonal elements of W in the singular value decomposition of $J = UWV^T$.

3.2. *Examples of ∀-Automata*

With automata, timed dynamic behaviors can be specified. Here is a set of examples for specifying performance using timed ∀-automata, as shown in Figure 1:

1. Let G be a constraint that the distance between the robot and its desired position is less than some constant value. Then Figure 1(a) specifies that the robot will eventually arrive its desired position.
2. Let G be a constraint that the error of a learning algorithm is less than a desired tolerance. Then Figure 1(a) specifies that the learning will eventually convergence. If let the state of ¬G in Figure 1(a) as a timed state with time bound t, it further specifies that the learning will be done within time t.
3. Let G be a constraint that the distance between the robot and obstacles is less than some constant value. Then Figure 1(b) specifies that the robot will never hit any obstacle. If G denotes that the current memory usage is out of the limit, Figure 1(b) specifies that the memory usage at any time is within its limit.
4. Let E be an external stimuli and F be a response. Then Figure 1(c) specifies that there is a response after stimuli within bounded time. Figure 1(d) specifies that such a response is within 5 time units.

Even though timed ∀-automata are powerful, still they are not able to represent all forms of performance metrics. For example, optimal performance over time $\min\int f(t)dt$ is not specifiable with timed ∀-automata. This form is mostly used for characterizing energy, efficiency or overall errors. Furthermore, specification with probability behaviors are not included either. However, it is not hard to add probability, for example, instead of "all runs" must be accepting and satisfying time constraints, we can say "x% runs" must be accepting and satisfying time constraints.

3.3 *Performance Comparisons*

Note that requirements specification defines what the system should do, rather than defining how the system is organized, i.e., its architecture. For example, behavior-based control [4,6] (which is arbitration based or a horizontal hierarchy) has a different form of architecture from function-based control [5] (which is abstraction-based or a vertical hierarchy); model-based systems have a different form of architecture from learning-based systems, event-driven systems have a different kind of architecture from time-driven systems. Different systems with different architectures can still be compared based on the behavioral interface under the formal performance specification. For example, given a set of requirements specification Rs and system A satisfies a subset As $\subseteq$ Rs and system B satisfies a subset Bs $\subseteq$ Rs. If As $\subseteq$ Bs, system A is not better than system B with respect to requirements Rs. Similarly, if system A satisfies requirement α and system B satisfies requirement β and if α implies β, system A is better than system B with respect to the requirement.

However, this specification does not define metrics on architectures. The measurement of performance should come from the customer's point of view, but the measurement of architecture should come from the developer's point of view, i.e., design time, debug time, upgrading time, modularity and the percentage of re-usable components.

4. SYSTEM VERIFICATION

For most dynamic systems, stability or convergence is the most important property that needs to be verified. For example, we can verify that equation $dx/dt = 0$ satisfies the property of $\forall$-automaton in Figure 1(a) with G as $|x|\leq\varepsilon$ for any positive number ε. The most commonly used method for the verification of such properties is the use of Liaponov functions. We developed a formal method based on model-checking, that generalizes Liaponov functions [13,17]. This method is automatic if the domain of interest is finite discrete and time is discrete [13].

The details of the model-checking method are out of the scope of this paper. The basic principle is to first find a set of invariants, each associated with an automaton-state in the timed $\forall$-automaton. Then, find a set of Liaponov functions, which are non-increasing in stable states and decreasing in bad states. Finally, find a set of local and global timing functions, where local timing functions are decreasing in timed states and global timing functions, like Liaponov functions, are non-increasing in stable states and decreasing in bad states, in addition to be bounded in values.

5. RELATED WORK AND CONCLUSION

Much work has been done in formal approaches to system specification and verification [1,2,7,8]. In general, there are two schools. One is to develop a uniform specification for both systems and their requirements; the other is to use two different specifications, one for systems and one for requirements. The advantage of the former is that the same formal approach can apply to both system synthesis and system verification. However, in most cases, if the specification language is powerful for both systems and requirements, the synthesis or verification tasks become hard. We advocate the latter approach, i.e., using timed $\forall$-automata for requirements specification and using Constraint Nets [13,18,19] for system modeling. Control synthesis [13,14] and verification [13,15,16,17,20] are also studied in this framework.

In this paper, we have shown how to use formal methods to specify the performance metrics of intelligent systems, with timed $\forall$-automata as an example. The advantage of formal methods over other methods lies in their precision and generality. Timed $\forall$-automata, with its graphical depiction and constraint specification, is a simple yet powerful formalism for specifying many properties of dynamic systems.

6. REFERENCES

[1] Alur, R., C. Courcoubetis, T.A Henzinger and P. Ho, "Hybrid automata: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, LNCS 736, Springer-Verlag, 1993, pp. 209 – 229.

[2] Alur, R. and D. Dill, "Automata for modeling real-time systems," M.S. Peterson, editor, ICALP90: *Automata, Languages and Programming*, LNCS 443, Springer-Verlag, 1990, pp. 322 – 335.

[4] Albus, J.S., "Outline for a Theory of Intelligence," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, pp. 473 – 509, May/June 1991.

[5] Arkin, R.C., *Behavior-Based Robotics*, The MIT Press, Cambridge, MA, 1998.

[6] Brooks, R.A., "Intelligence without reason," in IJCAI 1991, Sydney, Australia, pp. 569 – 595.

[7] Henzinger, T.A., Z. Manna and A. Pnueli, "Timed transition systems," J.W. deBakker, C. Huizing, W.P. dePoever, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, LNCS 600, Springer-Verlag, 1991, pp. 226 – 251.

[8] Manna, Z. and A. Pnueli, "Specification and verification of concurrent programs by $\forall$-automata," in Proc. 14th Ann. ACM Symp. On Principles of Programming Languages, 1987, pp. 1-12.

[9] Newell, A., "The Knowledge Level," *Artificial Intelligence*, 18(1), pp. 87-127, 1982.

[10] Newell, A., and Simon, H., GPS: A Program that Simulates Human Thought," Feigenbaum and Feldman,

editors, *Computers and Thought*, McGraw-Hill, New York, 1963.

[11] Sahota, M. and A. K. Mackworth, “Can situated robots play soccer?” in Proc. Artificial Intelligence, 1994, Banff, Alberta, pp. 249 – 254.

[12] Turing, A. "Computing Machinery and Intelligence." *Mind 59*, pp. 433-460, 1950. Reprinted in Feigenbaum and Feldman, editors, *Computers and Thought*, McGraw-Hill, New York, 1963.

[13] Zhang, Y., “A Foundation for the Design and Analysis of Robotic Systems and Behaviors”, PhD Thesis, University of British Columbia, Canada, 1994.

[14] Zhang, Y. and A. K. Mackworth, “Synthesis of Hybrid Constraint-Based Controllers,” P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems and Automatic Control*, LNCS 999, Springer-Verlag, 1994, pp. 552 – 567.

[15] Zhang, Y. and A. K. Mackworth “Specification and Verification of Constraint-Based Dynamic Systems,” A. Borning, editor, *Principles and Practice of Constraint Programming*, LNCS 874, Springer-Verlag, 1994, pp. 229 – 242.

[16] Zhang, Y. and A. K. Mackworth, “Will The Robot Do The Right Thing,” in Proc. Artificial Intelligence, 1994, Banff, Alberta, pp. 255 – 262.

[17] Zhang, Y. and A. K. Mackworth, “Specification and Verification of Hybrid Dynamic Systems Using Timed ∀-Automata,” *Verification and Control of Hybrid Systems*, LNCS 1066, Springer-Verlag, 1995.

[18] Zhang, Y. and A. K. Mackworth, “Constraint Programming in Constraint Nets”, V. Saraswat and P. Van Hentenryck, editors, *Principles and Practice of Constraint Programming*, MIT Press, Cambridge, MA, 1995, pp. 49 – 68.

[19] Zhang, Y. and A. K. Mackworth, “Constraint Nets: A Semantic Model for Hybrid Dynamic Systems,” *Journal of Theoretical Computer Science*, Vol. 138, No. 1, pp. 211 – 239, 1995.

[20] Zhang, Y. and Alan K. Mackworth, “Modeling and Analysis of Hybrid Systems: An Elevator Case Study,” H.Levesque and F.Pirri, editors, *Logic Foundations for Cognitive Agents*, Springer, Berlin, 1999, pp. 370-396.

Metrics for Intelligence: the Perspective from Software Agents
PRELIMINARY NOTES

Workshop on Performance Metrics for Intelligent Systems
National Institute of Standards and Technology

Line Pouchard
Collaborative Technologies Research Center
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6414
pouchardlc@ornl.gov

Each scientific development that claims to provide a "new way" for approaching existing problems needs proper (i.e. formal and quantifiable) evaluation methods and consensus-based criteria for measuring the validity of its claims. Taken together, these methods and criteria constitute the metrics by which new developments are being measured against their claims. Various claims have been made in the literature for the technology of intelligent software agents. Such claims include a new approach to programming providing a breakthrough comparable to the one achieved through object-oriented methods; an approach to programming that is more readily understood by non-programmers; an approach that lowers the costs of software inter-operability.

Software agents need proper metrics if the technology is to fulfill its promises and make a lasting impact. One characteristic distinguishing software agents from software developed with object-oriented and procedural methodologies is the anthropomorphic characteristics that agents exhibit. Various taxonomies for software agents currently exist [1, 2, 3]. Agents typically present one or several of the following characteristics:

- Pro-activeness and goal-orientation
- Reactiveness (reactive agents)
- Autonomy (rational agents, and others)
- Mobility (mobile agents)
- Learning and reasoning ability (deliberative agents, and others)
- Social ability: communication and cooperation (multi-agent systems)

An agent is considered intelligent if it can learn from its environment and modify its behaviors and goals to respond to environmental constraints that were uncertain and unforeseen at the time of development. Agents are thus particularly adapted to model environments where software components act autonomously on users' behalf and problem-solving environments where parameters of computation dynamically change during processing. The ability to learn for an agent is coupled with the ability to perform resource and knowledge discovery. This action may take the form of querying and updating knowledge-based systems. Knowledge discovery and interpretation bring latency to the agent and may impair the achievement of its overall goals. For instance, reactive agents that need a quick response time may not embody much learning and reasoning because the overhead renders the agent useless.

Software agents present one or some capabilities that are affected by the choice of specific components described in the Tools of Intelligence (see White paper). For instance, searching for a required object within a scene is one area where software agents have successfully been implemented. If you take the "scene" to be an information space like the Internet, information-gathering and retrieval agents display this capability and have been successful at performing the task. Deliberative agents such as Belief-Desire-Intention (BDI) agents exhibit the capability of remembering scenes and experiences as their Beliefs are based on this capability. These agents are also able to interpret and respond to unforeseen situations.

Agents' ability to autonomously execute processes on remote systems, given the appropriate permissions, is also a characteristic some intelligent systems (but not all) need to efficiently and effectively perform. This requires proper measures. This characteristic, known as mobility, has very different meaning for physical agents.

Mobility requires intelligence for software agents because true mobility requires resource discovery. For those agents designed as mobile agents the degree of mobility can constitute a measure of its intelligence. Mobile agents travel over networks such as the Internet and execute processes on remote platforms. Mobile agents may start execute a process on a particular machine, be unexpectedly interrupted, travel to another available platform, and continue the execution of the process from where it was interrupted. Such a mobile agent needs intelligence to interrupt and restart its execution autonomously without resetting, and for determining which resources to use in a networked environment. Network agents used for telecommunication applications (such as testing the reliability of a network) exemplify these types of agents.

Social intelligence needs to be measured in multi-agent systems. The degree of social interaction and the agents' ability to exhibit social behavior constitute an important criterion for multi-agent systems. Not all agent-based systems need to exhibit this characteristic (mobile agents may never need to talk to each other for instance). The type of social interaction between agents conditions knowledge acquisition and interpretation. The social model affects the individual pursuit of goals and may ultimately affect the survival of the system [4]. When one considers a multi-agent systems, there are at least two models. Both types of multi-agent systems, collaborative and cooperative, display the characteristics of open systems.

- Model 1: Each individual agent's goal is subservient to an over-arching goal of the system. We have a cooperative system, where agents agree not to pursue goals detrimental to each other and the whole system, even if these "careless" goals are in accordance with the individual agent's goal.

- Model 2: Each agent acts on its own behalf without recognizing a higher agent-entity with the ability to regulate its goals (there is still a need for a kind of supervisor agent that regulates communication). We have a collaborative system. This is the case for so-called rational agents, used especially in e-commerce, where agents act in a market-like environment, with the ability to bid for money on the goods and services each offers.

Agent-communication languages should theoretically let heterogeneous agents communicate, but none currently do [5]. A significant part of the inter-operability issue is the lack of a shared content language and ontology. An ontology expresses, for a particular domain, the set of terms, entities, objects, classes and the relationships between them with formal definitions and axioms that constraint the interpretation of these terms [6]. These definitions and axioms are written in a variety of logical languages (e.g. KIF [7]), and provide a formal theoretical basis to domain taxonomy. They can serve to automatically infer translation engines between software applications. By making explicit the implicit definitions and relations of classes, objects, and entities, ontologies also contribute to knowledge sharing and re-use across systems. The use of ontologies in agent-based systems is proposed as a criterion for the metrics of intelligent software agents. The degree of completeness and consistency of ontologies can be formally proven and provide a quantifiable criterion.

Ontologies constitute an important criterion for the metrics of intelligent software agents, in particular for agents exhibiting the social abilities of communication and cooperation. Software agents require the use of or a translation to a shared terminology and syntax in order to efficiently and effectively inter-operate. Agent-communication languages such as KQML meet the challenges of inter-operability with mitigated success [8]. Agent communication languages specify the possible use of ontologies in their syntax but do not require it. FIPA ACL proposes an ontology service as a normative specification [9].

In conclusion, software agents exist either as standalone or in social systems. Agents are made of components, and an agent-oriented architecture typically includes the agent application as well as an environment in which agents execute. They may execute on a single machine, on several machines connected locally or by wide-area network. These agents need a degree of mobility. They may be developed by different developers on different platforms, and therefore need a common communication language including protocol and ontologies (see [10] for an assessment of the state-of-the-art in this area). In addition, since agents may exhibit any combination of the characteristics above, some taxonomies of agents prefer a classification based on the domains in which software agents have been successfully implemented [11], rather than on their inherent characteristics.

Software agents also exist as whole, where an agent-based system is made of the agent and the underlying environment. The environment may include the knowledge repositories and ontologies which are key to the agents' degree of intelligence. For this reason, the mind/body dichotomy, and the proposition to measure the intelligence of the system based on the intelligence of the mind (controller), do not hold for agent based systems.

In addition to characteristics applicable to Constructed Systems with Autonomy, the metrics of intelligence for software agents need to include the following (not all these characteristics need apply for the same system):

- be domain-specific
- measure the degree of mobility
- present an agent communication language
- refer to ontologies.

References

1.Brenner, Walter; Zarnekow, Rudiger, and Wittig, Hartmut. *Intelligent Software Agents. Foundations and Applications*. Berlin: Springer Verlag; 1998; pp. 37-41.

2.Nwana, H. et. Al. "What is an agent?" Available at: http://www.labs.bt.com/projects/agents/publish/papers/review2.htm#agent

3.Wooldridge, M. Intelligent Agents. In *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. Weiss, Gehrard, ed. Cambridge, Mass.: MIT Press; 1999; pp.27-73.

4.Huhns, Michael N. and Stephens, Larry. Multiagent Systems and Societies of Agents. In *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. Weiss, Gehrard, ed. Cambridge, Mass.: MIT Press; 1999; pp. 79-120.

5.Singh, Munindar P. Agent communications languages: rethinking the principles. *IEEE Computer.* 1998; 31(12):40-47.

6.Gómez-Pérez, Asuncion. Knowledge Sharing and Reuse. In *The Handbook of Expert Systems*. Boca Raton, FL: CRC Press; 1998; pp. 10/1-10/36.

7.Knowledge Interchange Format. Available at http://logic.stanford.edu/kif/dpans.html.

8.Labrou, Yannis; Finin, Tim, and Peng, Yun. Agent communication languages: the current landscape. *IEEE Intelligent Systems & Their Applications*. 1999; 14(2):45-52.

9. Foundation for Physical Intelligent Agents (FIPA). Available at http://drogo.cselt.it/fipa and http://www.fipa.org.

10. Wooldridge, M. and Jennings, N.R. Applications of Intelligent Agents. In *Agent Technology: Foundations, Applications, and Markets*. Berlin: Springer Verlag, 1998; pp. 3-28.

11. Nwana, Hyacinth S. and Ndumu, Divine T. A perspective on software agents research. *The Knowledge Engineering Review*. 1999; 14(2):125-142.

Minimal Representation Size Metrics for Intelligent Robotic Systems

Dr. Arthur C. Sanderson
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
sandea@rpi.edu

ABSTRACT

The minimal representation size criterion provides a metric for the configurational complexity of robotic tasks and may be used to evaluate alternative algorithms, strategies, and architectures for the accomplishment of specific tasks. The principles of explicit and implict representation are used to define this complexity and the resulting information measure derived may be considered as a measure of configurational intelligence of the system. Specifically, these measures indicate the internal explicit information required to specify the accessible states of the robotic system using its available perception and actuation capabilities. The resulting approach may be used to evaluate and guide applications tasks such as robotic assembly and multisensor manipulation.

Keywords: *minimal representation size, intelligent systems, performance metrics, robotics*

1. INTRODUCTION

Intelligent robotic systems couple computational intelligence to the physical world and such systems may be considered as intelligent agents that perceive the environment, and select an action or sequence of actions to affect the environment. Such an intelligent agent constructs an internal "representation" of the environment, and uses reasoning to choose among alternative actions. Specifically, we can define robots as "active, artificial, intelligent agents whose environment is the physical world". Such agents may be distinguished from software agents, human agents, and others.

Such an intelligent robot is regarded as "rational" if the agent makes decisions to choose actions that accomplish a known task goal, or increase a performance measure of the task. It is important to distinguish the presence of intelligence from the metric of performance. Intelligence (reasoning), in itself, does not maximize overall performance. However, intelligence may be used to choose among a set of candidate actions that may improve performance or achieve a goal.

An intelligent robot may also be characterized by its autonomy. In the context of these definitions, autonomy refers to the capacity of the robot to define its own goals or sub goals, often based on its perception and internal representation of the environment. Autonomy widens the scope of tasks, which the same system can perform without reprogramming, but in general, requires more sophistication in the design and architecture of the system. The non-autonomous system may accomplish a smaller set of tasks and may require efforts to constrain or redesign the environment to conform to task assumptions.

The structure of an intelligent robot agent includes perception, representation, reasoning, and representation. The implementation of such an agent requires two major components: (1) Algorithms that define the representation structure and reasoning sequence, and (2) Architecture that defines the organization of the system to accomplish set goals and performance. In practice, the selection of the architecture has been strongly intertwined with the nature of the representation. For example, one simple intelligent robot defines a perception-action pair such as "move hand if you touch the hot stove!" Such a reflex action might be expressed as a look-up table in which state representation is a simple binary element.

As the complexity of robots and tasks increases, a single reflex action is inadequate to create required behaviors, and architectural approaches have tended to evolve in two directions. First, *hierarchical* architectures have been based on the definition of a hierarchy of *explicit* representation of the robot state. A hierarchy of perceptual representation may involve image features, shapes, objects, scenes, etc., while a hierarchy of actions may involve joint motion, arm motion, robot motion, sensor-based motion etc. The formal definition of such a hierarchical architecture [1] has provided an important basis for building consistent, predictable, and programmable robotic systems.

A second trend has been the development of *behavioral* architectures [3] that expand upon simple reflexes by creating a network of interdependent reflexes in order to increase the sophistication of the behaviors. One such behavioral approach is the *subsumption* architecture [5] that utilizes finite state machines to impose a priority setting logic on the reflex actions. The nature of such behavioral architectures is to incorporate an *implicit* representation of the environment in order to define a simplified state space of perceptions and actions. From a systems perspective, the behavioral architecture utilizes constraints or assumptions about the environment to identify a subspace (manifold) within the explicit state space. A reflex action, or set of actions, may then be defined within the subspace with the logical consistency to achieve goals and performance metrics.

The distinction between *explicit* and *implicit* representations is important to the interpretation of intelligence in systems. A simple task example helps to illustrate these distinctions. Consider a room with a single door containing a mobile robot. The robot task goal is to exit the room, and it may have a performance metric of minimum time to exit. Several different types of algorithms may be considered:

(1). Random search (Figure 1a)

The robot moves in random directions without using perception, mechanically bouncing off the walls. Eventually, it is guaranteed to exit the room.

(2). Wall following – simple reflex (Figure 1b)

The robot uses a simple sensor to detect presence or absence of an adjacent wall. The algorithm:

IF ('wall-is-in-front') THEN ('Turn-Right') ELSE ('Follow-wall-on-left')

is guaranteed to find the door, though the path may be long.

(3). Perception - Explicit state representation (Figure 1c)

The robot uses a sophisticated vision sensor to view the door, acquire a perception, P, update the global internal state representation, GS, and plan an explicit path to the door.

(4). Perception – Implicit state representation (Figure 1d)

The robot defines an implicit mapping of GS to local state, LS, that is consistent with the desired goal state. By mapping perception into LS, rather the GS, the resulting algorithm is often more efficient and simpler to implement. In this case, consider a sensor that perceives only the width, W, of the door, but no other attributes of the environment. We choose W to be the local state representation, LS = W, and define a local reflex algorithm to choose an action, A:

Choose A to increase W.

(a). If robot, R, moves toward the door, W' > W.

(b). If R moves perpendicular to the door, then W'>W.

The resulting *local* changes in W move the robot toward and through the door, achieving the global goal. However, LS is never sufficient to explicitly locate the robot in the room, i.e. determine GS. This strategy is analogous to a potential field mapping related to the perceived door width feature of the room. The same strategy may be used as a feature-based method to guide a peg-in-hole or other assembly problem using visual servoing of the area of the target hole [26].

These examples illustrate several types of tradeoffs in the design of intelligent systems, and also confirm that the most intelligent system may not result in the optimal performance on a given task, as illustrated in the performance of the feature-based example. First, for this purely geometric task, we can define one component of the intelligence of the system, the *configurational complexity* as the *information required to represent the accessible states of the internal representation of the system.* "Accessible states" are defined as those states that may be achieved as goal states of the system through its perception-action algorithms. In this sense, the representational intelligence of the system is equated to the size of the internal representation space.

For the examples in Figure (1), the configurational complexity is found to be: (a). 1 bit, (b). 3 bits, (c). 30 bits, and (d). 10 bits, where a resolution of 10 bits has been assumed for the vision sensor used in (c) and (d). By considering the approximate number of steps required to achieve the result, on can similarly compute the cumulative complexity for each of the tasks to be: (a). 100 bits, (b). 75 bits, (c). 60 bits, and (d). 20 bits. Therefore, the minimal complexity approach to the task is given by strategy (c) and may be regarded as a tradeoff between explicit and implicit information needed for the task.

In addition, the time (number of steps) required for each task is implicit in the cumulative information and reflects the inherent deficiencies in the worst case scenarios for (a) and (b). Based on the viewpoint of encoded residuals discussed in the next section, one can also calculate the encoded implicit information for each strategy: (a). 20 bits, (b). 18 bits, (c). 0 bits, (d). 12 bits.

Figures (e) and (f) emphasize the inherent assumptions that are often present in such systems. Strategies (a) and (b) are not guaranteed to succeed for problems (e) and (f), where the subspace manifold defined by the strategy is no longer guaranteed to contain the goal. Strategies (c) and (d) may still succeed but require more steps and a more sophisticated algorithm.

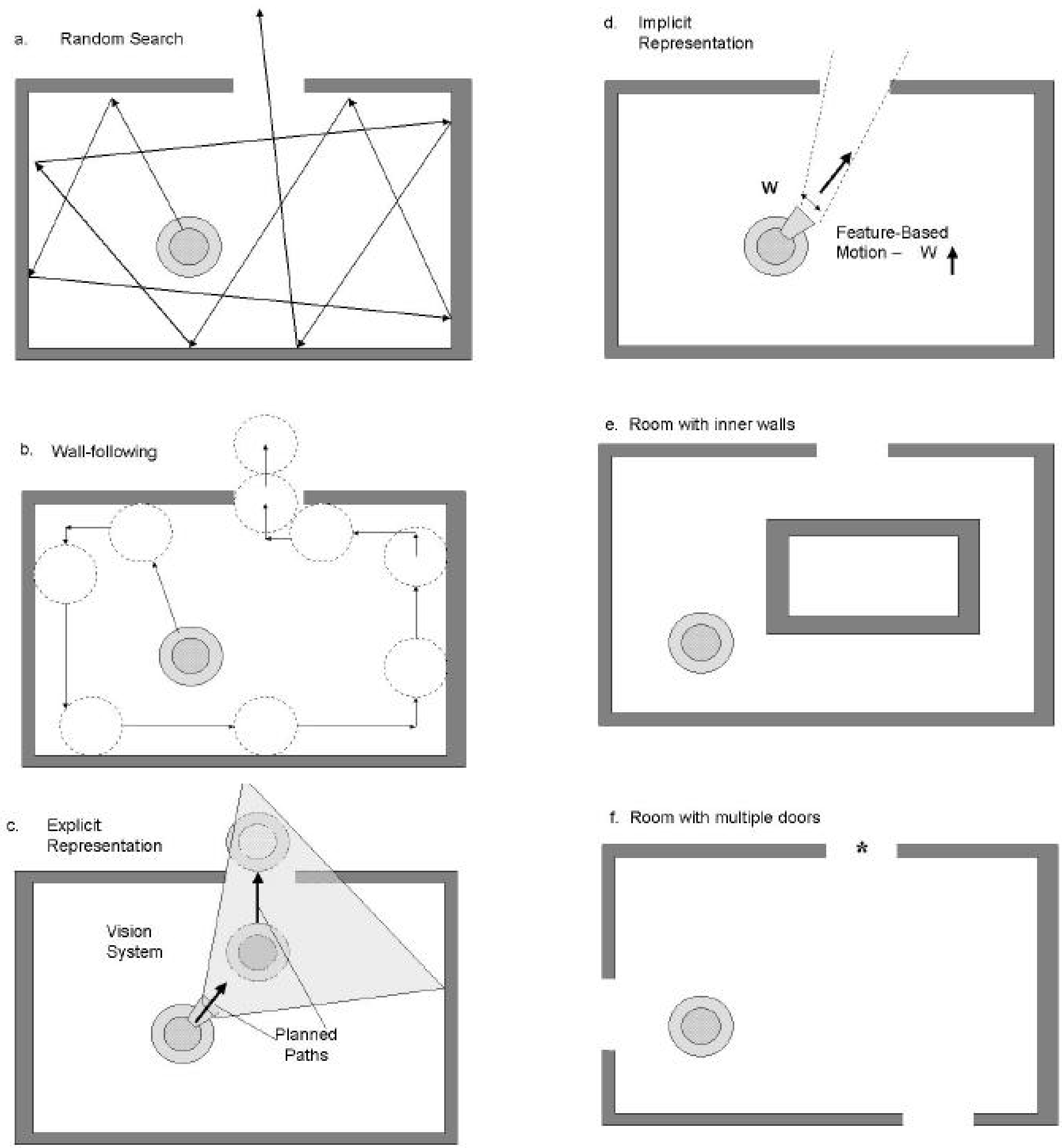


Figure (1). Examples of alternative strategies for the task of exiting a room through the door: (a). Random search, (b). Wall-following, (c). Explicit representation and global planning, (d). Implicit representation and local reasoning. All four strategies will accomplish the basic task. However, (a) and (b) are not general and will fail when the environment differs from the basic assumptions, such as in (e) with inner walls, and in (f) with multiple doorways.

2. MINIMAL REPRESENTATION SIZE

The minimal representation size (MRS) methods [6,18,19,23,24] used in this work are also called "minimum description length" methods in the literature. The MRS approach introduces an information measure of model complexity and has been applied to a number of related problems in attributed image matching [22], shape matching [11], density estimation [4], and model based sensor fusion [11-17]. The minimal representation criterion defines the minimal overall data representation among a choice of alternative models and trades off between the size of the model (e.g. number of parameters) and the representation size of the encoded residuals. Intuitively, the smaller, less complex, representation is chosen as the preferred model for a given performance criterion. In terms of the robotic systems we consider here, the representation size combining state and model information serves as a measure of system intelligence, and the MRS criterion will select the minimal complexity system for a given task performance. In practice, the MRS criterion has advantages in the attainment of consistent metrics without the introduction of problem specific heuristics or arbitrary weighting factors. The MRS family of methods provides a type of "universal yardstick" for data and models from disparate sources, and therefore has been successfully used in multisensor fusion interpretation problems.

The MRS criterion has been proposed as a general criterion for model inference by Rissanen [19] and by Segen and Sanderson [23]. It is an expression of the ideas on algorithmic information theory pioneered by Solomonoff [24], Kolmogorov [18], and Chaitin [6]. The MRS approach is based on the principle of building the shortest length program that reconstructs observed data. The length of this program or *representation size* depends on both the statistics of the sensors and on the systems "knowledge" of the environment, specified by a set of models and constraints.

More formally, the *representation size* is the length of a program in bits that, when executed on a deterministic Universal Turing Machine (UTM) [7] would reproduce the observed data on the output tape. A model based encoding scheme is used in which the data is thought to be arising from one of the several available models in a model library, Q. The models may differ in structure and number of parameters. The observed data D is encoded by specifying an instantiated model q and the deviations or *residuals* of the data D from the selected model $q \in Q$. The resulting representation size is

$$L[q,D|Q] = L[q|Q] + L[D|q,Q]$$

$$= L[q|Q] + L[A|q,Q] + L[D|Q,q,Q]$$

where L[q,D] is the total representation size of data D when explained using model q, given a model library Q. L[d|A,q,Q] is the number of bits needed to encode the data deviations or residuals from the model, given a coding algorithm, A. L[A|q,Q] is the number of bits required to specify the coding algorithm itself, given an environment model. L[q|Q] is the number of bits required to encode the environment model (structure and parameters) given a model library, Q.

According to the minimal representation principle, the best explanation of the observed data is the one with the smallest representation size

$$Q_{opt} = \arg\min_{q \in Q} L[q|Q] + L[A|q,Q] + L[D|A,q,Q].$$

This approach finds the simplest explanation of the data that is most likely, and objectively trades off between model size, algorithm complexity, and observation errors. Rissanen [19] showed that a finite set of random samples from a class of probability distributions would be complexity bounds as defined by Kolmogorov [18] and others [6,24], and the representation size can be used to choose among alternative distribution models. Barron and Cover [4] showed that such a minimal representation size probability distribution is statistically accurate and the rate of convergence is comparable to other methods of parametric and nonparametric estimation. In our previous work [13-17], we have structured the model-based pose estimation problem such that the pose transformation parameters are isolated elements of the statistical model, and may be estimated by the minimal representation criterion.

3. PARTS ENTROPY AND INFORMATION MEASURES FOR ASSEMBLY

Geometric task complexity is directly related to the geometric state space and the precision of state definition or partitioning. In earlier work [20], we have defined the *parts entropy* as a measure of configuration uncertainty in mechanical systems with particular application to assembly analysis and assembly planning. In this formulation, the entropy of a distribution of independent objects, or parts, is given by

$$H_n = H_n (P_1, \ldots, P_n) = - \Sigma P_k \log_2 P_k .$$

where uncertainty in position and orientation is described by the joint probability distribution $P(x,y,z,\alpha,\beta,\chi)$ over the joint ensemble. As an entropy measure [7], H may also be interpreted as the information required to specify the position of the objects in their geometric configuration space.

The part entropy of an object is defined with respect to the mechanically distinguishable positions and orientations, and the resolution, d, in each coordinate degree of freedom. The symmetry of an object therefore strongly affects the resulting orientational entropy and is defined by the set of group operations that leave the object invariant. For example, a sphere has 0 bits of orientational entropy, while a cube with 10 bits of resolution would have 24 bits of entropy.

The part entropy may be used as a basis for the configurational representation size, and is directly related to the set of constraints or other geometric assumptions made on the environment. For example, a flat surface reduces the entropy of parts that sit on it. The entropy of a cube sitting on a table (with 10 bits of resolution) is 28 bits, while a general rectangular solid will be 30.1 bits, and a cylinder may vary from 20 to 30 bits depending on its proportions.

For an assembly task, we consider a set of parts $\{Q_i\}$, I = 1,…,N, such that the part relationships are defined by join probabilities $P[Q_{1\ldots}Q_N]$, and the parts entropy is defined as the joint entropy $H[Q_{1\ldots}Q_N]$. If the parts are positioned independently, for example, prior to assembly, then the probabilities will be independent:

$$P[Q_{1\ldots}Q_N] = P(Q_1)\,P(Q_2)\ldots P(Q_N),$$

and

$$H[Q_{1\ldots}Q_N] = \Sigma\, H(Q_i).$$

As the assembly task proceeds, individual parts entropies decrease as parts are positioned, and the entropy of the ensemble decreases as part dependence is increased during mating operations. In this sense, an overall goal of the assembly task is to reduce the joint entropy of the ensemble of parts. If we define the entropy of the final rigid assembly to a reference frame with $H_Q = 0$, then the relative entropy of parts and subassemblies may be tracked as a function of time and the entropy flow of the process described in terms of bits per second, that is, information flow. Alternative systems choices and parts designs may be compared in terms of the entropy flow and used to guide decisions on assembly system design. An example described in [20] tracks the parts entropy sequence for sequential assembly for three different electronics assembly strategies. Similar concepts of part probability distributions may be linked to tolerance specifications of assemblies, and have been used to evaluate assemblability based on maximum likelihood methods [21], and used to guide assembly planning tasks [8-10].

4. MULTISENSOR FUSION MANIPULATION EXAMPLE

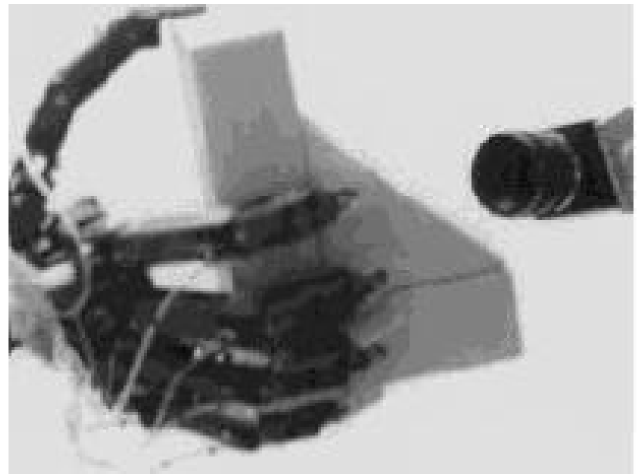


Figure (2). Five fingered anthropomorphic robot hand manipulating an object. The camera observes motions and minimal representation metrics are used to determine object configuration [16].

The MRS approach has been applied to the problem of multisensor fusion for pose identification of objects using in manipulation by a robot hand. The setting of the task is shown in Figure (2). A five-fingered Anthrobot-3 [2] hand is mounted on a six degree-of-freedom (DOF) articulate PUMA-760 robot arm. The hand is provided with finger tip tactile sensors that sense planar surface contact with the grasped object. The hand is in the field of view of a calibrated camera with edge detection algorithms. A polyhedral object is grasped by the hand and manipulated within the camera view.

In this task scenario, the minimal representation criterion is used to integrate the perception and manipulation steps through the use of consistent information-based criterion for consistency of interpretation of the manipulation with the viewed object pose from the camera. In this task, both the camera information and the tactile sensing data is extremely noisy and uncertain.

The minimal representation formulation of this problem is described in detail in [16]. In this approach, the model-based representation of the hand-eye coordination is described by a set of general constraint equations

$$h(y;z) = 0$$

where Y is a set of model features, and Z is a set of observed data features. In general, such constraints may themselves depend on other model features. Often observed data features may not be related to actual events and identified as unmodeled data features.

The association between the observed data features and the model features is defined by a correspondence w, and this correspondence is a part of the identified model. In addition, a model of the feature extractor, F, for vision and tactile sensing is used to described the process. Application of the MRS approach defines a representation size for each candidate model and set of observations subject to the *data constraint manifold*, DCM, defined by h(y;z). The representation size of the model and encoded residuals is minimized within the measurement subspace locally orthogonal to the DCM.

In general, the search over many candidate models and correspondences is difficult and does not lend itself to linear continuous search techniques. In [16] we use a differential evolutionary algorithm [25] to carry out this search and identify viable interpretations as minimal representation size interpretations of manipulation and sensing states of the system. Figure (3) shows an example of the evolution of the configuration states of the system as the differential evolutionary algorithm proceeds. The system converges to a well-defined and consistent interpretation of the current state (figure (4)).

5. DISCUSSION

The minimal representation size criterion provides a metric for the configurational complexity of robotic tasks and may be used to evaluate alternative algorithms, strategies, and architectures for the accomplishment of specific tasks. The principles of explicit and implict representation are used to define this complexity and the resulting information measures derived may be considered as a measure of configurational intelligence of the system. Specifically, these measures indicate the internal explicit information required to specify the accessible states of the robotic systems using its available perception and actuation capabilities. The resulting approach may be used to evaluate and guide applications tasks such as robotic assembly and multisensor manipulation.

As discussed here, the characterization of tasks is defined with respect to geometric configurations. An important extension of this work is to consider the application of such a formulation to a more general task space involving, for example, force and dynamics of the system requirements.

Figure (3). Differential evolution algorithm utilizes representation size metric to search for consistent interpretations of object pose in the hand of manipulator. The minimal representation size pose requires the minimum information to represent.

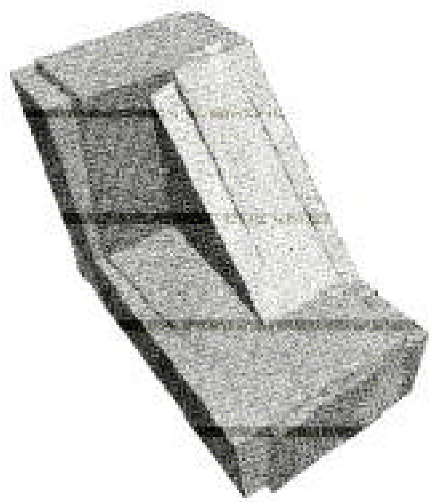


Figure (4). Final minimal representation pose of the object determined by the differential evolution search.

A second extension of this work is the consideration of intelligent robotic systems with adaptation and learning capabilities. As shown in the multisensor fusion manipulation example, the representation size may be used as a criterion for evolutionary learning of configuration interpretations. In general, this approach might be used to guide learning of algorithmic structure and strategies leading to more sophisticated behaviors.

REFERENCES

[1]. Albus, J.S., H.G. McCain, and R. Lumia, “NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)”, National Bureau of Standards Report SS-GSFC-0027, March 13, 1987.

[2].Ali, M.S., and C. Engler, “System Description document for the Anthrobot-2: A dexterous robot hand,: Tech. Memo 104535, NASA Goddard Space Flight Center, <D, March, 1991.

[3]. Arkin, R. C., *Behavior-Based Robotics (Intelligent Robots and Autonomous Agents)*, MIT Press, May, 1998.

[4]. Barron, A. R., and T. M. Cover, “Minimum complexity density estimation”, *IEEE Trans. Inform. Theory,* vol. 37, pp. 1034-1054, July, 1991.

[5]. Brooks, R. A., “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, vol. RA-2, No. 1, March, 1986.

[6]. Chaitin, G. J., “A theory of program size formally identical to information theory,” *J. ACM*, vol. 22, no. 3, pp. 329-340, 1975.

[7]. Cover, T. M., and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[8]. Homem de Mello, L. S. and A. C. Sanderson, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 228-240, April 1991.

[9]. Homem de Mello, L. S. and A. C. Sanderson, "Representation of Mechanical Assembly Sequences," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 211-227, April 1991.

[10]. Homem de Mello, L. S., and A. C. Sanderson, "Two Criteria for the Selection of Assembly Plans: Maximizing the Flexibility of Sequencing the Assembly Tasks and Minimizing the Assembly Time Through Parallel Execution of Assembly Tasks", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 5, pp. 211-227, October 1991

[11]. Joshi, R., and A. C. Sanderson, "Model-based Multisensor Data Fusion: A Minimal Representation Approach", *1994 IEEE International Conference on Robotics and Automation,* San Diego, CA, May, 1994.

[12]. Joshi, R., and A. C. Sanderson, "Multisensor Fusion and Unknown Statistics using the Minimal Representation Criterion", *Proceedings 1995 IEEE International Conference on Robotics and Automation*, May, 1995.

[13]. Joshi, R., and A. C. Sanderson, ``Multisensor Fusion and Model Selection using a Minimal Re presentation Size Framework", *Proceedings IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, December 8-11, Washington, D. C., USA, Dec. 1996.

[14]. Joshi, R., and A. C. Sanderson, “Experiments in Tactile/Visual Multisensor Fusion”, 1997 IEEE International Conference on Intelligent Robotics, Monterey, CA, July, 1997.

[15]. Joshi, R., and A. C. Sanderson, "Minimal Representation multisensor fusion using differential evolution", *Proc. 1997 Int. Symp. Computational Intelligence in Robotics and Automation,* Monterey, CA., July, 1997

[16]. Joshi, R., and A. C. Sanderson, "Minimal Representation Multisensor Fusion using Differential Evolution", *IEEE Transactions on Systems, Man, and Cybernetics,* January, 1999, Vol. 29, No. 1, pp. 63-76

[17]. Joshi, R., and A. C. Sanderson, *Multisensor Fusion: A Minimal Representation Framework*, World Scientific Publishers, December, 1999.

[18]. Kolmogorov, A. N., "Logical basis of information theory and probability theory," *IEEE Trans. Inform. Theory*, vol. IT-14, no. 5, pp. 662-664. 1968.

[19]. Rissanen, J., "Modeling by shortest data description," *Automatica,* vol. 14, 465-471, 1978.

[20]. Sanderson, A. C., "Parts Entropy Methods for Robotic Assembly System Design", *Proceedings of the IEEE International Conference of Robotics,* Atlanta, GA, March 13-15, 1984.

[21]. Sanderson, A.C., "Assemblability Based on Maximum Likelihood Configuration of Tolerances", *IEEE Transactions on Robotics and Automation*, June, 1999, Vol. 15, No. 3, pp. 568-572.

[22]. Sanderson, A. C., and N. J. Foster, "Attributed Image Matching using a Minimal Representation Criterion", Invited Paper, *Proceedings of the AAAI Spring Symposium*, Palo Alto, California, March 27--29, 1990.

[23]. Segen, J., and A. C. Sanderson, "Model inference and pattern discovery by minimal representation methods," Tech Rept. CMU-RI-TR-82-2, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1981.

[24]. Solomonoff, R.J. " A formal theory of inductive inference," *Inform. Contr*., vol 7, pp. 1-22, 1964.

[25]. Storn, R., and K. Price, "Minimizing the real functions of the ICEC'96 context by differential evolution," in *Proc. 1996 IEEE Int. Conf. Evolutionary Computation,* May, 1996, pp. 842-844.

[26]. Weiss, L. E., A. C. Sanderson, and C. P. Newman, "Dynamic Sensor Based Control of Robots with Visual Feedback", *IEEE Journal of Robotics and Automation*, Vol. RA3, No. 5, pp. 404-417, October 1987.

Metrics for System Autonomy

Part I : Metrics Definition

Arie ("Arik") Yavnai
Head, Center for Autonomous Systems
RAFAEL, P.O.Box 2250 (39)
Haifa 31021, Israel
Tel: + 972-4-990-8801; Fax: + 972-4-990-8055
e-mail: arikyav@rafael.co.il

ABSTRACT

In many real world applications, system *Autonomy* is the most single significant and meaningful attribute of Intelligent Autonomous Systems - *IAS*. This paper presents performance metrics for *IAS*, which are related to *Autonomy*. Metrics are presented and defined. These metrics are currently being used in on-going research, development and engineering work.

1. INTRODUCTION

From an engineering point of view, performance metrics for *IAS* are needed for establishing and developing the following system level processes: a) a sub-process within the multi-phase system engineering process, e.g., system requirements analysis; b) preliminary and detailed design process; c) Concept-of- Operation development process; d) comparative evaluation of alternative designs.

A fundamental question which is related to *IAS* performance metrics is: Which entity is more meaningful and practical to define and to measure with respect to *IAS* performance – *Autonomy* or *Intelligence*? Our position is that from the user point of view, as well as from the system architect and designer point of view, *Autonomy* is the premier characteristic attribute of an *IAS*. Although *Intelligence* enables *Autonomy*, it is not considered by us as either an appropriate or a practical system design objective or a system performance requirement *per se*.

The concept of *Autonomy* is probably more meaningful, more communicatable, and more precisely measureable, and it is easier to come to a consensus about what *Autonomy* or what an *Autonomous System* is all about, rather than what is *Intelligence* or what is an *Intelligent System*.

2. AUTONOMY

Currently, two distinguished approaches to define system autonomy are used by researchers and groups within the intelligent autonomous systems (including autonomous agents) community. The first approach defines autonomy as an entity which is assigned to the subject system or to the subject agent by a higher level authority, e.g., a supervisor agent. Within the context of this approach, autonomy is defined with respect to the assigned responsibility of a system or an agent. Within this context, autonomy reflects the agent's decision-making capability and authority, and the degree of self control the agent has over its own decisions, see [1]. This approach is more commonly used within the autonomous agents community. The other approach defines system or agent autonomy with respect to its self capability to accomplish its assigned mission goals while operating under uncertain dynamic environment, uncertain dynamic scenario and self faulty situations, and without or with very little human or external agent intervention, [2], [3]. We are using the later approach.

Definition: *Autonomy* is an attribute of a system which characterized its ability to accomplish the system's assigned mission goals without any or with only minimal external intervention, while operating under constraints and under uncertain dynamic environment and scenario conditions.

3. CRITERIA FOR METRICS

In the sequel, some guidelines for metrics selection are proposed.

3.1 *Scope*

The proposed metrics should reflect system autonomy as perceived by an external observer. Therefore, the autonomy should be measured outside the system boundary, i.e., in the interface of the system with external entities. Figure 1, in the sequel, illustrates the context of Autonomy Evaluation, as perceived by an external observer. Four entities are identified within the relevant context, namely: a) a Remote user or supervisor; b) an External Agent; c) Environment & Scenario; d) System Under Evaluation (SUE), which is the Autonomous Intelligent System to be evaluated.

3.2 *Autonomy Relevance*

Meaningful, effective, and measurable metrics for system autonomy should reflect the influence of the following factors as related to system autonomy:

- Level of Abstraction of the commands and the data provided to the autonomous system by the remote user/ supervisor or by an external agent.
- Information bandwidth between a remote user/ supervisor or an external agent, and the system under evaluation.
- The levels of complexity, dynamics and uncertainty which are attributes to the environment under which the system is operating and executing its mission.
- The levels of complexity, dynamics and uncertainty which are attributes to the system operating scenario while executing its mission.

3.3 *Generality*

Although the meaning of performance metrics is usually domain and application specific, more general entities, such as the principle of *entropy* can be used within the framework of *IAS* performance evaluation. In our work, *entropy* is used as a general measure of entity uncertainty, and is applied to measure various parameters. Using *entropy* as a general tool for representing uncertainty in the domain of control and system engineering was proposed by Saridis [4].

3.4 *Structure Independence*

The metrics for Autonomy should be independent of the internal structure, e.g. : a) number of levels of the hierarchy; b) the decomposition of *IAS* internal processes to resolution scales; c) the computational paradigms, e.g. fuzzy vs. neural networks, and d) other internal specific features. The attempt to establish metrics which takes into account internal specifics of the system will lead to an endless confusing and unpractical effort, and to unstable solution-depended metrics. System Autonomy is a system attribute as perceived by an external observer. In analogy, consider a consumer which want to buy a new car. His decision will not depend on whether the fuel injection control system uses a fuzzy logic based controller or a differential geometry based non-linear controller. However, his decision will probably be based on user-centered parameters such as: fuel consumption (kilometers per liter), number of passengers, safety measures, to name but a few. In such evaluation, the internal specifics are irrelevant. So are the internal specifics when one has to evaluate the performance of an Autonomous Intelligent System.

4. METRICS

In the following section, the metrics used for *IAS* performance evaluation are defined. The nomenclature used is described as follows:

4.1 *Nomenclature*

(1)

Nomenclature :

ChS - *Channel Sensitivity*

EnS - *Environment Sensitivity*

InS - *Information Sensitivity*

ScS - *Scenario Sensitivity*

H - *Entropy*

$H(\Psi)$ - *System Entropy*

$H(\Gamma)$ - *Environment Entropy*

$H(\Lambda)$ - *Scenario Entropy*

C - *Channel Capacity of Data Link between Remote-User or External Agent to System*

Ψ - *System Under Evaluation (SUE)*

Γ - *Environment*

Λ - *Scenario*

I - *Externally provided system Information (global and mission related)*

Φ - *Remote User*

Ω - *Problem Context*

n - *Time step index*

4.2 *Entropy*

We are using *entropy* as a measure of uncertainty of system state, environment state, or scenario state. The uncertainty associated with predicting the next entity state, given the current entity state, is a measure of the entity irregularity or 'disorder'. The less is the entity regularity, the greater is the next state prediction uncertainty and the greater is the associated entropy. Thus, entropy can be used as a measure of environment uncertainty as well as a measure of scenario uncertainty. Entropy can also be used as a measure of system uncertainty, which is directly related to system performance. It can represents the uncertainty in selecting the appropriate control from the set of all admissible controls [4]. Entropy can also be used for representing performance, e.g., system tracking error along a planned trajectory in the system state space.

We define *entropy* as follows:

(2)

Entropy Definition

$$P(X,n,l) = Prob\ \{X(n+1)=X_l \mid X(n)\} \ ;$$

$$X_l \in \{X\}$$

$$H(X,n) = -\sum_l P(X,n,l) \bullet \ln P(X,n,l)$$

X - *Entity State* - (*e.g., best control action; Environment State; Scenario State*)

H - *Entropy*

$H(\Psi)$ - *System Entropy*

$H(\Gamma)$ - *Environment Entropy*

$H(\Lambda)$ - *Scenario Entropy*

4.3 Channel Sensitivity

Channel Sensitivity- ChS, is defined as the differential change of the system entropy which results after a differential change in the channel capacity of the information data link between a remote-user and the System Under Evaluation - *SUE*, or between an external agent and the *SUE*, has occurred.

(3)

Channel Sensitivity :

$$(ChS)_n = \frac{\Delta H(\Psi,n))/H(\Psi,n)}{\Delta C(n)/C(n)} \quad || \ \Psi=\Psi_m;\ C=C_f;\ \Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{ChS} = \frac{1}{n}\sum_{k=1}^{n}(ChS)_n$$

$$\Psi_\mu \in \{\Psi\};\ X_\phi \in \{X\};\ \Gamma_\alpha \in \{\Gamma\};$$

$$\Lambda_\delta \in \{\Lambda\};\ \Phi_\beta \in \{\Phi\};\ \Omega = (\Gamma,\ \Lambda,\ \Phi, I)$$

Definitions:

If $\overline{ChS} \prec 0 \Rightarrow$ *SUE is Non-Autonomous w.r.t.* C, *under context* Ω

If $\overline{ChS} \equiv 0 \Rightarrow$ *SUE is Autonomous w.r.t.* C, *under context* Ω

If $\overline{ChS} \succ 0 \Rightarrow$ *SUE is Non-Supervisable w.r.t.* C, *under context* Ω

4.4 Environment Sensitivity

Environment Sensitivity- EnS, is defined as the differential change of the system entropy which results after a differential change in the environment entropy, or uncertainty, has occurred.

(4)

Environment Sensitivity :

$$(EnS)_n = \frac{\Delta H(\Psi,n)/H(\Psi,n)}{\Delta H(\Gamma,n)/H(\Gamma,n)} \quad || \ \Psi=\Psi_m;\ C=C_f;\Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{EnS} = \frac{1}{n}\sum_{k=1}^{n}(EnS)_n$$

$$\Psi_m \in \{\Psi\};\ C_f \in \{C\};\ \Gamma_a \in \{\Gamma\};$$

$$\Lambda_d \in \{\Lambda\};\ \Phi_b \in \{\Phi\};\ \Omega = (\Lambda,\ \Phi, C, I)$$

Definitions:

If $\overline{EnS} \succ 1 \Rightarrow$ *SUE is Non-Autonomous w.r.t.* Γ, *under context* Ω

If $0 \prec \overline{EnS} \leq 1 \Rightarrow$ *SUE is Partly Autonomous w.r.t.* Γ, *under context* Ω

If $\overline{EnS} \equiv 0 \Rightarrow$ *SUE is Completely Autonomous w.r.t.* Γ, *under context* Ω

4.5 Scenario Sensitivity

Scenario Sensitivity- ScS, is defined as the differential change of the system entropy which results after a differential change in the scenario entropy, or uncertainty, has occurred.

(5)

Scenario Sensitivity :

$$(ScS)_n = \frac{\Delta H(\Psi,n)/H(\Psi,n)}{\Delta H(\Lambda,n)/H(\Lambda,n)}$$

$$\|\ \Psi=\Psi_m;\ C=C_f;\Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{ScS} = \frac{1}{n}\sum_{k=1}^{n}(ScS)_n$$

$$\Psi_m \in \{\Psi\};\ C_f \in \{C\};\ \Gamma_a \in \{\Gamma\};$$

$$\Lambda_d \in \{\Lambda\};\ \Phi_b \in \{\Phi\};\ \Omega = (\Gamma,\ \Phi, C, I)$$

Definitions:

If $\overline{ScS} \succ 1 \Rightarrow$ SUE is Non-Autonomous w.r.t. Λ, under context Ω

If $0 \prec \overline{ScS} \le 1 \Rightarrow$ SUE is Partly Autonomous w.r.t. Λ, under context Ω

If $\overline{ScS} \equiv 0 \Rightarrow$ SUE is Completely Autonomous w.r.t. Λ, under context Ω

4.6 *Information Sensitivity*

Information Sensitivity- InS, is defined as the differential change of the system entropy which results after a differential change in the system global and mission related externally provided information, has occurred. The information includes the Mission Plan and the related Data Bases which provided to the autonomous system by the remote user/ supervisor or by an external agent, prior to mission execution, or while the mission is executed.

(6)

Information Sensitivity :

$$(InS)_n = \frac{\Delta H(\Psi,n))/H(\Psi,n)}{\Delta I/I}$$

$$\|\ \Psi=\Psi_m;\ C=C_f;\ \Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{InS} = \frac{1}{n}\sum_{k=1}^{n}(InS)_n$$

$$\Psi_\mu \in \{\Psi\};\ X_\phi \in \{X\};\ \Gamma_\alpha \in \{\Gamma\};$$

$$\Lambda_\delta \in \{\Lambda\};\ \Phi_\beta \in \{\Phi\};\ \Omega = (\Gamma,\ \Lambda,\ \Phi,\ C)$$

Definitions:

If $\overline{InS} \succ 1 \Rightarrow$ SUE is Non-Autonomous w.r.t. I, under context Ω

If $0 \prec \overline{InS} \le 1 \Rightarrow$ SUE is Partly Autonomous w.r.t. I, under context Ω

If $\overline{InS} \equiv 0 \Rightarrow$ SUE is Completely Autonomous w.r.t. I, under context Ω

4.7 *Adaptation Rate Sensitivity*

Adaptation Rate Sensitivity - ARS, is defined as the differential change of the system entropy rate which results after a differential change in the entropy of the subject entity, e.g., environment or scenario, or uncertainty, has occurred. Similarly, Adaptation Rate Sensitivity can be defined in relation with differential changes of channel capacity or information.

(7)

Adaptation Rate Sensitivity:

$$(ARS)_n = \frac{\Delta(\partial H(\Psi,n)/\partial n)/(\partial H(\Psi,n)/\partial n)}{\Delta H(X,n)/H(X,n)}$$

$$\|\ \Psi=\Psi_m;\ C=C_f;\Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{ARS} = \frac{1}{n}\sum_{k=1}^{n}(AdR)_n$$

$$\Psi_m \in \{\Psi\};\ \ C_f \in \{C\};\ \Gamma_a \in \{\Gamma\};$$

$$\Lambda_d \in \{\Lambda\};\ \Phi_b \in \{\Phi\};\ \Omega = (\Gamma,\ \Phi, C, I)$$

Definitions:

If $\overline{ARS} \succ 1 \ \Rightarrow$ *SUE is Non-Autonomous w.r.t. X, under context* Ω

If $0 \prec \overline{ARS} \leq 1 \ \Rightarrow$ *SUE is Partly Autonomous w.r.t. X, under context* Ω

If $ARS \equiv 0 \ \Rightarrow$ *SUE is Completely Autonomous w.r.t. X, under context* Ω

5. SUMMARY

Metrics for system autonomy has been defined and presented. Following the metrics, a specific measure for a certain application can be derived directly. Associated with each definition, the broad classification of the SUE was defined.

REFERENCES

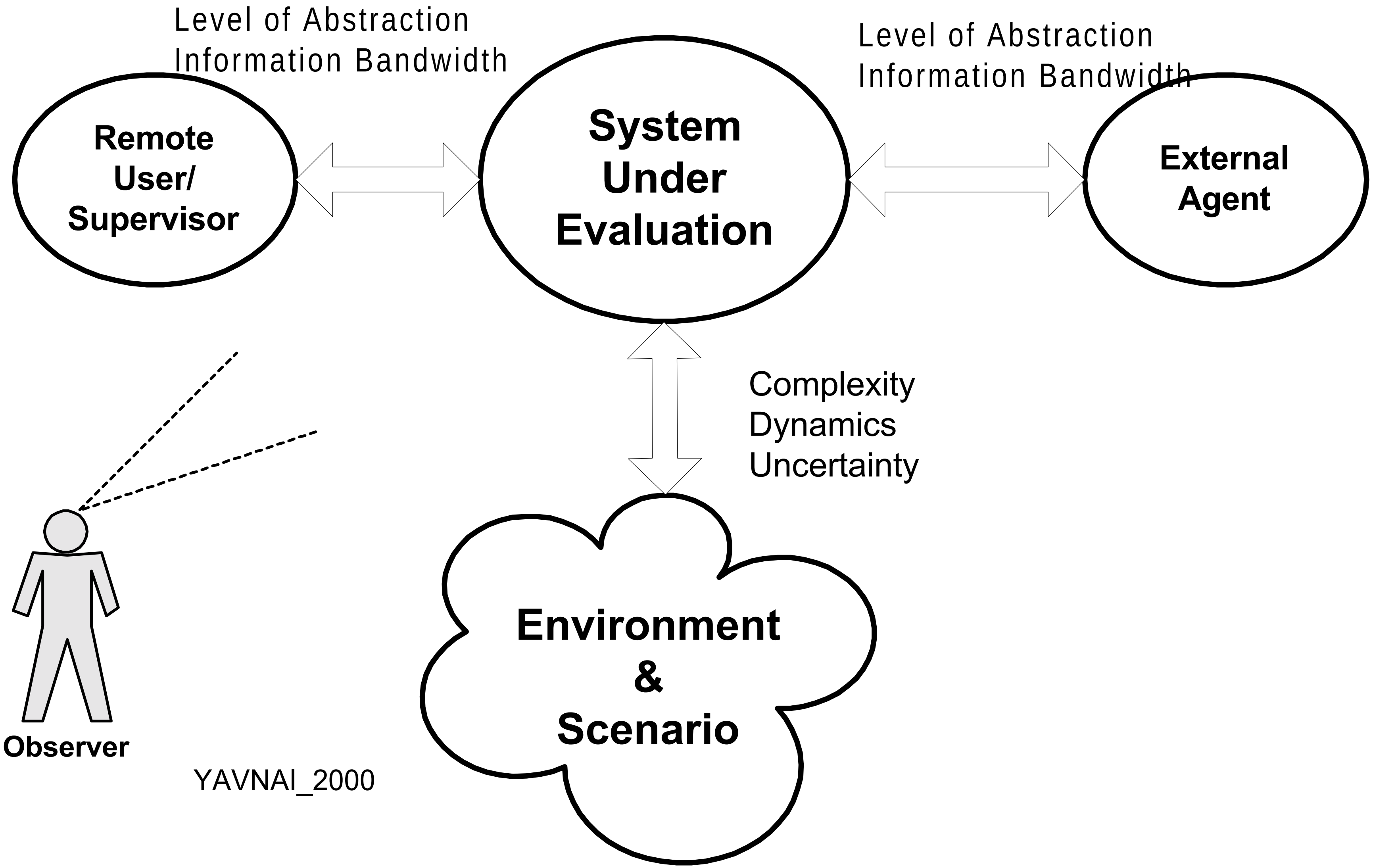
[1] Barber, K.S., (1999), "Agent Autonomy: Specification, Measurement, and Dynamic Adjustment", 1999, pp. 1-8.

[2] Yavnai, A., (1989), "Criteria of System Autonomability", Proc. Intl' 2nd Conference on Intelligence Autonomous Systems, Amsterdam, Dec. 1989, pp. 448-458.

[3] Yavnai, A., (1991), "Entropy-based Criteria for Iintelligent Autonomous Systems", Proc. IEEE Intl' Symp. Intelligent Control 1991, Arlington VA., August 1991, pp. 55-60

[4] Saridis G. N. (1995), "Stochastic Processes, Estimation and Control - The Entropy Approach", Wiley-Interscience, New-York, NY, 1995.

CONTEXT OF AUTONOMY EVALUATION



In Defense of the Additive Form for Evaluating the Multidimensional Vector

Dr. Leonid M. Polyakov
Globe Institute of Technology
Leonid@globeinstitute.org

ABSTRACT

The topic of this discussion is an artificial (not a natural human) intelligence measurement. It would be better to call it an evaluation rather than a measurement. The Additive Evaluation Method is the **only real** method to make a evaluation of the vector value.

KEYWORD: *intelligence, measurement, expert. additive.*

ADDITIVE FORM.

Artificial Intelligence, like a human one, is a composition of the different additive abilities such as reasoning, learning, decision-making, object recognition, and so on. The multifunctional nature of intelligence can be represented as a **vector**.

The intelligence measurement is not the same as a multiobjective optimization of the intelligence systems. There are many different methods of optimization (Preference Structures, Compromise Approach, Lexicographic Ordering Approach, Genetic Approach, Pareto approach, etc.) [4,5, and other]. All of these methods work with each function of the intelligence separately and determine preferences and a system's rank, but not an intelligence value. The additive function is presented in the most of the research works [2,3,6,7,9-14, and other].

The measurement is a process of **assigning numbers** to the objects or events in accordance with certain rules of the system. The number assignment is possible just on the scalar scale. There are three types of axioms related to a measurement process: identity axioms, rank axioms, additivity axioms. These axioms determine four scale levels: scale of names, range scale, interval scale and ratio scale. The analyses of these scales are done in [2]. Only additivity axioms can be applied to the real measurement. These axioms can be applied just to the **scalar** scale, as it was mentioned above. A vector doesn't meet these conditions. Just, the weighted-sum approach and utility functions can be used in this case [3,7] as the method of multivariable scales aggregation and converts vector into a sufficient scalar.

The last question is how to determine the value of weight. The most known and usable method is an expert method, but there are several **analytical methods** to find out the value of this function [2,6].

Opponents of these methods of the aggregation function complain against the application of a human expertise as a source of information. They dispute an expert ability to produce objective information. Yes, a collective expertise has an element of subjectivism but today we don't have a better way to measure a vector's values to make a comparison of two or more vectors' values. Is this, a wonderful fact in that we use an expert's intellectual ability in the intelligence measurement? Certainly not, because the intelligence can be measured by the scale of the intelligence. Only a human being has the best sense of the value of the intelligence functions. Each separate intelligence function can be measured by appropriate methods but, as an integrated value, intelligence has to be presented as a scalar.

There are many different methods to measure each separate intellectual ability. For example, the value of the ability to learn can be presented as a ratio of an increment of intelligence to an increment of information. The number of iterations, or the number of rules and trials (trial and error method), or the entropy method, etc can determine the value of information. So, the learning ability is:

$$L = d(I)/d(If). \qquad (1)$$

The amount of new information available to the different systems can change the intelligence value of these systems.

A values of a separate intellectual abilities (variables) don't give any ideas about artificial intelligence integrated value. Aggregation of the separate variables can be done on the base of the utility theory. The utility of intelligent alternative can be presented as [2]:

$$U_A = \sum_{i=1}^{n} U_i \qquad (2)$$

where *Ui* is an utility of *i-th* basic variable,
n is a number of variables.

From (2) [2], we can get the quality index of j-th alternative (domain specific by design) in nondimensional units

$$Q_j = \sum_{i=1}^{n} W_i (F_i)*(Fi/Fi\ max) \quad (3)$$

Where *Wi (Fi)* is a weight function of *i-th* variable *(Fi)*.

A set of variables has to be named for each problem separately.
Usually one of the variables is an investment value of the j-th alternative (C_j). In this case, equation (4) can be rewritten as:

$$Q_{j*}(C\ max/ Wc) = \sum_{i=1}^{n-1} [Wi (Fi)/Wc] * C_{max} *(Fi/Fimax) - C_j. \quad (4)$$

This equation presents the evaluation of j-th alternative measured in cost units (dollars). Now we can use money as a real universal scale of the measurement. Some opponents can say, "it is immoral". A measurement is not a moral category! C_j can be added to the left and the right parts of the equation (4). In this case we can get the value of $Q_{j*}(C\ max/ Wc)$ presented in dollar units. This value includes only intelligence variables and can be called the intelligence value of the j-th
alternative

$$I_j = \sum_{i=1}^{n-1} [Wi (Fi)/Wc] * C_{max} *(Fi/Fimax). \qquad (5)$$

Where *Wc* is a weight function of variable *Cmax*.

This is the direct way to calculate **profit (**political factors are included). It is **one more reason** to use the Utility Method and scalar scale. **No other method permits us to get an intelligence evaluation in dollar units.** Each time in the shopping center, when we are buying something we use ours preferences and convert a vector value into a scalar value presented in the dollar units.

The intelligence measurement is not a new problem. The famous IQ and WAIS-3 [8] tests are the possible ways to make an evaluation of the human intelligence. These tests present an aggregated value of the multifunctional intelligence and convert a vector value into a scalar value.

The opponents to these testes pointed out to the possible social problems bounded to these methodic. In case of artificial intelligence measurment this problem does not make sense.

Conclusion.

The Additive Evaluation Method is the **only real** method to make a evaluation of the vector value. It can't be write off from the tools of intelligence value evaluation. Artificial intelligence of the system should be measured and presented as scalar.

This method is the **only one,** which can gives financial evaluation of artificial intelligence application.

Contemporary artificial intelligence systems are design as a domain-oriented systems. **Only the expert** can determine the importance of each intellectual function with regard to the certain domains.

References:

1. Pareto M. D'economic Politque. Paris 1927, 695p.

2. Polyakov L.M., Kheruntsev P., Shklovsky B., Elements of the Automated design of the electrical automated equipments of Machine tools. Publ. "Machinostrojenie", Moscow, 1974, 157p.

3. John Von Newmann. O. Morgenstern, Economic Behavior and Theory of Games, 1944, 650p.

4. Dhar, V. Stein, R. Intelligent Decision Support Methods Prentice Hall, 1997, 244p.

4. Mitsuo Gen, Runwei Cheng. Genetic Algorithms Engineering & Optimization. A Wiley-Interscience Publication, 2000, 490p.

5. Sigford Y., Pazvin R., Project PATTERN: a methodology for determining relevance in complex decision making. IEEE Transaction. Eng. Manag. V.EM-12, No.1, 1965, 210p.

6. Fishburn P. C., Additive Utilities with Incomplete Product Set: Applications to priorities and Assignments- Operations Research, V. 15, 1967, No.3. p.537-542.

8. Charles G. Morris, Albert A. Maisto, Psychology. Prentice Hall, 1999, 682p.

9. Marino Y. P. Technological forecasting for decision making, American Elsevior Company Inc. N.Y., 1972.

10. Fishburn P. C. A Study of Independence in Multivariate Utility Theory. Econometric, 37, No.1, 1969, p.7-121

11. Fishburn P.C. Independence in Utility Theory with Whole Product Sets. Operations Research, V.13, 1965, p.28-45.

12. Schziver A. Forecast Air Review, V. 16, No.3, 1965, p.12-23.

13. Larichev O. The Art and Science of Decision Making, "Nauka", Moscow, 1979, 196p.

14. Hall A. Experience of Methodology for Large Engineering Systems, "Soviet Radio," M, 1975, 120p.

A Method For Evaluating the "IQ" Of Intelligent Systems

Robert Finkelstein

University Of Management & Technology
1925 N. Lynn Street, 3rd Floor
Arlington, Virginia 22209
(703)-516-0035 Ext. 24
And
Robotic Technology Incorporated
11424 Palatine Drive
Potomac, Maryland 20854-1451
(301)-983-4194
RobertFinkelstein@compuserve.com

ABSTRACT

This paper describes a **pragmatic** process measuring the "IQ" of individual intelligent robots and groups of intelligent robots. We offer definitions for the characterizations "intelligence" and "IQ." We define **metrics and submetrics** for individual robots and the collective, using the Analytic Hierarchy Process (AHP) to calculate weights for the metrics and submetrics. They can then be used to evaluate alternative technologies and systems for achieving individual and collective intelligent behavior in robots.

The defined metrics and submetrics for **individual robots** include:

- **Intelligence** (decomposed into the ability to make Correct Decisions and Right Decisions, and to Learn)
- **Effectiveness** (decomposed into the ability to achieve Objectives, Goals, and Priorities)
- **Efficiency** (decomposed into Accuracy, Precision, Time efficiency, Energy Efficiency, and Side Effects)

The defined metrics and submetrics for **groups of robots** include:

- **Command And Control** (decomposed into Leadership, Followership, and Efficiency)
- **Communications** (decomposed into Message Initiation, Transmission, Understanding, and Efficiency)
- **Effectiveness** (decomposed into the ability to achieve Objectives, Goals, and Priorities)

The values of the metrics are determined by experimenting with designed robotic systems, in the context of a scenario, in a simulation or field experiment. The weighted metrics can be combined to obtained an "IQ" score for individuals or the collective.

Keywords: *Intelligent Robots, Metrics, IQ, Intelligent Systems, AHP, Robot Groups*

1. Introduction

A little learning is a dangerous thing;
Drink deep, or taste not the Pierian spring;
There shallow draughts intoxicate the brain
And drinking largely sobers us again.
----- Alexander Pope

The brain is the most overrated organ.
----- Woody Allen

There are no satisfactory definitions of *human* intelligence, so it is not surprising that are no generally accepted definitions of *machine* intelligence either. One implicit definition, "the ability to cope with the unexpected, and the ability to bring about the unexpected," is from a comment in the *Economist* about the major attribute for a good U.S. president. Another suitable definition of intelligence, is **"the ability to make an appropriate choice or decision."** The intelligence need not be at the human level. The ability to make an *appropriate* choice is common to all long-time survivors, including roaches and rats. *Appropriate*, for organisms, usually means enhancing the ability to reproduce, the primal goal in life. A chicken is an egg's way of making another egg. *Appropriate* intelligence for a robot might mean the ability to accomplish its mission under a variety of conditions.

It has been difficult to measure human intelligence in a satisfactory way since the first "IQ" (Intelligence Quotient) tests were developed at the start of the 20th century. The tests, and their interpretations, remain controversial. The measurement of machine intelligence, however, is a somewhat easier task primarily because (1) the functional domains of interest are more narrowly defined than for people and (2) the underlying mechanisms of machine intelligence are more accessible to experimentation than the means of human intelligence.

When intelligent machines are designed for a limited set of functions - such as performing search and rescue of people trapped in collapsed buildings - they can be tested within that small sphere of endeavor regardless of the intended level of their intelligence (e.g., whether they are as intelligent as insects or humans). In this sense, the robot's IQ test is analogous to a person's aptitude test for a job or profession. More importantly: for measuring the "IQ" of an intelligent machine, the tester has access to the underlying intelligent control system. This allows the intelligent control system to be connected to an avatar of the machine in a simulated environment. The tester can know the ground truth - have a "god's-eye view" - of everything in the environment, including every external manifestation of behavior by the robot avatar as well as the control system's every internal state (including learned and adaptive behavior). The intelligent control system need not know - or care - that it is controlling an avatar in a simulation and not a physical system in the real world. Of course, the validity of the simulation is only as good as the ability of the underlying model to replicate the real-world environment of interest.

Intelligence can be decomposed into the ability to make a *correct* decision (the optimum decision given complete knowledge), a *right* decision (the optimum decision given limited knowledge), and *learning* (the ability to adapt to the environment, without necessarily making a decision which leads *immediately* to altered, observable external behavior). While *intelligence* is an important metric (measure of merit) for an autonomous intelligent robot, there are two other key measures (as per Peter Drucker): *efficiency* (a measure of how well the autonomous robot does things right) and *effectiveness* (a measure of how well the robot does the right thing). These metrics take into account other system variables and characteristics, including: energy expenditure, mobility, reliability, stealthiness, etc. An intelligent robot with a failed engine or damaged servo motors cannot move to accomplish its mission no matter how well it has planned its path. Some researchers are redefining human IQ to include a variety of human talents, including physical skills. Indeed, some anthropologists believe that human intelligence was quite fragmented and narrowly focused task by task (as in *Homo neanderthalensis*) until recently, when intelligence became synthesized in *Homo sapiens sapiens*. Likewise, the "IQ" of an intelligent robot might include the **combined metrics of cognitive and physical abilities**: Intelligence, Effectiveness, and Efficiency. However they are labeled or amalgamated, these metrics can be quantified and used to test the performance - the "IQ" - of any autonomous intelligent system.

1.1 Sundry Definitions Of Intelligence

"Civilization advances by extending the number of important operations which we can perform without thinking."
--- Alfred North Whitehead

For organisms, intelligence is a pragmatic mechanism of survival; and all measures of intelligence (whether for organism, man or machine; whether genetically encoded, pre-programmed, or learned) involve an **ability to make appropriate selections** [1, 2], **choices, or decisions**. Human intelligence involves "the degree to which an individual can successfully respond to new situations or problems. It is based on the individual's knowledge level and the ability to appropriately manipulate and reformulate that knowledge (and incoming data) as required by the situation or problem," [3]. Intelligence can be identified by an ability to cope with the unexpected and an ability to bring about the unexpected, abilities against which to judge presidents, among other notables, over history [4]. The subjective word "appropriate," in relating intelligence to "appropriate" choice, implies that a system can be intelligent only in relation to a defined goal or environment.

Intelligence requires an ability to use information (where information, according to Claude Shannon, is that which reduces uncertainty) [2], and using information includes an ability to detect new, non-chance associations [5]. Chen defines intelligence (individual or organizational) "as the attainment of relevant goals in specified contexts using appropriate means and resulting in positive outcomes," [6], which is the same as saying, as above, "intelligence is the ability to make an appropriate choice."

A behaviorist would say, in the spirit of the Turing Test, that if humans, machines, or organizations (collectives) **behave** intelligently, then they are; if they manifest consciousness, then they are conscious. Two parts of intelligence are: (1) epistemological, in which the world is so represented that solutions to problems follow from the facts expressed in the representation; and (2) heuristic, in which there is the mechanisms that solves the problem and selects actions on the basis of information (most work in artificial intelligence is devoted to the heuristic part) [7]. Entities can place different emphasis on these two kinds of mechanisms of intelligence, depending on the context.

In one view [8], attributes of systems with higher intelligence include:

- ❑ mental attitude (beliefs, intentions, desires);
- ❑ learning (ability to acquire new knowledge);
- ❑ problem solving;
- ❑ understanding (implications of knowledge);
- ❑ planning and predicting consequences of actions, comparing

alternative actions;

- knowing limits of knowledge and abilities;
- drawing distinctions between similar situations;
- synthesizing new concepts and ideas, acquiring and employing analogies, originality;
- generalizing;
- perceiving and modeling the external world;
- understanding and using language and symbolic tools.

Some hold that the last attribute - language - is the prime determinant of higher intelligence; that every representation of knowledge is an interpretation, not decision-making or expertise [9]. Symbolic manipulation [10] - communication - creates second order reality; an advanced, intelligent system must be able to perceive second order reality (the meaning and values attributed to first order reality) as well as first order reality (the reality accessible to perceptual consensus, physical reality) [11].

A machine with higher intelligence should be able to: adapt (changing itself or the environment) for survival; reason about its own organization, reasoning ability, and external domains; plan internal activities (database searches, decision-making) and external activities (sending messages, physical actions); select among decision-making processes; make decisions using values associated with possible actions; reason about its reasons for taking actions; value itself to avoid changing itself in a harmful way. Ideally, the system should be self-conscious as well as self-adaptive [12].

The higher intelligent system should possess meta-knowledge, i.e., it should have knowledge about what it knows without having to search exhaustively. For example, the system should know whether it has knowledge about grapefruit if asked the size of grapefruit [13]. Knowledge includes representations of facts, generalizations, and concepts, organized for future use [5]. Knowledge of general truths does not require a special metaphysically distinct ingredient in humans [14] - machines can be designed to know such truths. "Knowledge is more than a static encoding of facts; it also includes the ability to use those facts in interacting with the world ... knowledge of something is the ability to form a mental model that accurately represents the thing as well as the actions that can be performed by it and on it. Then by testing actions on the model, a person (or robot) can predict what is likely to happen in the real world," [15].

"The use or handling of knowledge" is cognition [16], "an intellectual process by which knowledge is gained about perceptions or ideas," [17]. An intelligent system can be designed to learn ("any deliberate or directed change in the knowledge structure of a system that allows it to perform better on later repetitions" of a task [18]). But it would be difficult to give it common sense, which involves a larger variety of different types of knowledge than expertise (a large amount of knowledge of relatively few varieties) [19]. A robot is *behaving* consciously if it [20]:

- receives information about its environment;
- recalls and compares past experiences;
- evaluates the quality of its experiences;
- makes conceptual models of its environment;
- projects consequences of alternative future actions;
- chooses and implements actions which further its goals.

By exhibiting purpose and intention, a machine would behave *as if* it had free will and the ability to choose [21].

1.2 Group Intelligence

Organizations or collectives can become intelligent through the emergent behavior of its organisms and machines. "Emergent behavior involves the repetitive application of seemingly simple rules that lead to complex overall behavior," [22]. The emergent behavior can be that of an ant and ant colony, a person and an organization, or a robotic vehicle and a combat platoon. Collective intelligence in insect societies, especially for certain of the ants, bees, and termites, is reasonably understood. "Higher forms of intelligence arise from the synchronized interaction of simpler units of intelligence," [23]. This is true as well of "higher" forms of life, such as dolphins, wolves, apes, and humans. Social intelligence allows an individual organism to "analyze and respond correctly (intelligently) to possible behavioral responses of other members of the group," [24]. Collective intelligence, an advanced form of intelligence, "involves group intelligence in which individuals submerge their individual identity" [24] to the group's responses to the threats and opportunities in the environment. Communication among individuals is essential for collective intelligence, whether by pheromone, vision, sound, touch, or email. Information technology is now affecting the collective intelligence and evolution of the human species, possible leading to the emergence of a global intelligence, a system of individual and collective humans and machines [25]. But, as always, the essence of intelligent behavior is control - at least *self-control*.

There have been a number of programs attempting to develop cooperative mobile robots, and over 200 papers have been published concerning mobile cooperative robots [26]. Cao, Fukunaga, and Kahng [27], on which much of the following discussion of cooperative behavior is based, define **collective behavior** generically as "any behavior of agents in a system having more than one agent," while **cooperative behavior** is defined as "a subclass of collective behavior that is characterized by cooperation." Cooperation should lead to the enhanced performance of the collective over that of the simple aggregation of individuals (i.e., the whole should be greater than the sum of its parts). Cao et al. cite the following definitions of cooperative behavior (from various sources):

- To associate with another or others for mutual, often economic, benefit.

- Joint collaborative behavior that is directed toward some goal in which there is a common interest or reward.
- A form of interaction, usually based on communication.
- Joining together for doing something that creates a progressive result, such as increased performance or saving time.
- Given some task specified by a designer, a multiple robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the mechanism of cooperation) there is an increase in the total utility of the system.

As posed by Cao et al., the fundamental issue is: *given a group of robots, an environment, and a task, how should cooperative behavior arise?*

The architecture of a computing system is that part which remains unchanged unless an external agent changes it. The group architecture of a cooperative system is the infrastructure on which collective behaviors are implemented and determines the abilities and constraints of the system. The group architecture for cooperative robots includes such considerations as: robot heterogeneity and homogeneity, the ability of each robot to recognize and model other robots, and communications. Also, the architecture must be able to avoid conflicts among robots for resources, such as paths through the environment, goal objects in the environment, and communications bandwidth.

In creating a group architecture, there are a number of alternative design decisions. The architecture may be centralized or decentralized. Centralized architectures are characterized by a single control agent. Decentralized architectures, which are prevalent, may be either distributed or hierarchical. In the former, all agents are equal with respect to control, while the latter are locally centralized. Decentralized architectures may lead to emergent properties of systems, such as intelligence or self-organization. Their inherent advantages over centralized architectures include fault tolerance, natural exploitation of parallel processes, reliability, and scalability. Most robot architectures hybrid, where, for example, a central planner exerts high-level control over mostly autonomous agents. A group of robots is homogeneous if the capabilities of the individual robots are identical; otherwise they are heterogeneous (forming a more complex system).

Cooperation among robots can arise from eusocial behavior (as opposed to explicit cooperative behavior) which results from the behavior of individuals and not necessarily an a priori effort at cooperation (e.g., ants and bees are eusocial). There are many sorts of self-organizing systems (in which there has been much research), but especially with respect to biological systems, whether individual organisms (in which the individuals parts are self-organizing) or social groups (human or otherwise). The aggregation of relatively limited individuals leads to the collective's more capable intelligence (this is true of human society as well). Individual robots that are selfish and utility-driven, but must cooperate in order to survive, will display emergent cooperative behavior. Explicit cooperation, as among humans, can be driven by a desire to maximize individual utility, so there are economic and game-theoretic approaches to examining cooperation.

It is difficult for human designers to account for the multiplicity of control variables and contingencies to achieve cooperative behavior in robots. It is easier to design the robots so that they learn to cooperated and adapt to their environment. A number of techniques are being developed for this approach, including the use of neural networks and genetic algorithms.

The robotic group may employ various types of communications processes for inter-agent interaction, including, in one taxonomy: interaction by means of the environment; interaction by sensing; interaction by explicit communications. The simplest, most limited type of interaction occurs when the environment itself is the communications medium, providing the equivalent of a shared memory among a group of robots. There is no explicit communication or interaction among the individuals.

Another form of group communications occurs when individuals sense and perceive one another without engaging in explicit communications. Using a suitable sensor (e.g., vision, acoustic, chemical, touch), the individuals must be able to distinguish members of the group from other entities in the environment. Resulting collective behavior includes flocking and pattern formation relative to neighboring individuals.

Higher-order tactical group behavior generally requires explicit communication among individuals, which can be directed (to known recipients) or broadcast (to unknown recipients). Architectures that enable this type of communication resemble communications networks, and communications protocols are necessary for inter-robot communications. The message carrier can consist of various portions of the electromagnetic spectrum (e.g., radio frequency, microwave, optical, infrared) or other transmission mechanisms (e.g., acoustic, chemical).

In order to function relative to others in a group, or with respect to predators (threats) and prey (targets), individual organisms (or robots) must be able to model the intentions, beliefs, actions, capabilities, and states of those others. The ability of individuals to model others in a group reduces the need for communications; it encompasses implicit communications via the environment and perception and includes representations of other individuals which can be used to make inferences about the actions of those individuals.

There are many prospective means of achieving cooperative behavior among robots. The most direct is to explicitly program the desired behavior. This is difficult and tedious in that the programmer must a priori account for all possible contingencies. Other methods are more promising, including biological (e..g, social insects) behavioral approaches, task decomposition and allocation approaches, game-theoretic approaches, machine learning approaches, and approaches based on cooperation as an emergent property of complex group dynamics. Geometric approaches include multi-agent path planning, moving to formation, and pattern generation.

For most military applications, explicit leader-follower relationships are important, especially where robotic forces will be integrated with conventional forces. These roles and abilities may exist in all of the robots, where leaders are anointed - or emerge - based on circumstance (as is often the case for people). Or leaders may be specially trained as such.

For example, group behavior to achieve coordinated movement in the world, such as path planning, can be centralized (with a leader or universal path planner making decisions) or distributed (with individual agents planning and adjusting their paths). They may be hybrids, combining on-line, off-line, centralized, and decentralized elements. Planning systems may take into account all robots, or plan the path of each one independently. Factors include dynamically-varying global and individual priorities, environmental constraints and obstacles, and the allocation of space-time resources. Conflicts may be resolved by a central manager or negotiated among individuals.

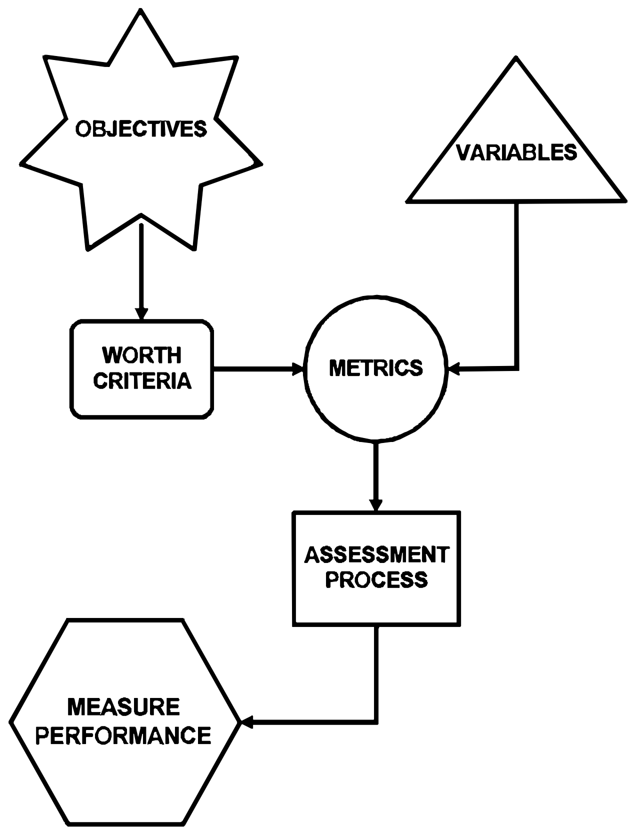
2.0 Evaluation Process

In order to evaluate the performance of an intelligent robot (or group of robots) we can employ a *pragmatic*, behaviorally-based, teleological, and functional approach to measuring its "IQ" as follows:

- Define the purpose or mission or objectives of the robot (or group of robots)
- Derive the worth criteria by which the robot's performance may be assessed
- Organize and integrate the worth criteria into a consistent assessment structure
- The assessment structure employs suitable variables (endogenous, status, and exogenous), metrics and submetrics, a means of weighting or ordering the metrics and submetrics, and a means of evaluating performance against the ordered metrics and submetrics.
- Measure the performance of the robot or group of robots, in the context of the desired scenario and environment, in a simulation or field exercise, calculating "IQ" from the evaluated, weighted metrics and submetrics.

The evaluation process is illustrated in **Figure 1**.

Figure 1. Evaluation Process



The worth criteria are specified as metrics (e.g., worth criteria which are measurable either objectively or subjectively), which are commonly labeled as measures of merit, measures of effectiveness, measures of efficiency, measures of performance, and so forth. Measures Of Merit (MOM) are often the worth criteria associated with the system as a whole, while Measures Of Performance (MOP) are worth criteria often associated with the system's subsystems (which may descend to the n^{th} subsystem level of the system). Using these labels, the MOP are below the MOM in a hierarchy of worth criteria, with the MOM comprising the MOP and being a function of the MOP values. For example, a robot's MOP may be "Time Efficiency," and this, along with other MOP, then compose the MOM "Vehicle Efficiency."

The objectives, for example, may be to demonstrate the intelligent and cooperative behavior on the part of multiple autonomous robots or robotic vehicles in the context of scenarios relevant to a class of military missions. The primary objectives of achieving (1) intelligent behavior, and (2) cooperative behavior, lead to the definition of worth criteria focused on two system levels: (1) the individual robot or robotic vehicle as a system, and (2) the group of robots or robotic vehicles as a system (i.e., a *system of systems*).

2.1 Procedural Difficulties

The evaluation procedure is a formal procedure, as opposed to an observer's using purely subjective judgment and intuition to pronounce the performance of the system to be a success or failure. However, formal decision-making procedures do not preclude (and often require) the use of subjective judgment. Subjective judgment must be used in developing worth criteria and assigning them to the various performance consequences, as well as in deriving relative weights for the worth criteria (i.e., trading off worth among the various criteria). But if subjective judgment is made explicit and logically consistent, then it can be examined and questioned by all interested parties. The result is more likely to be free of incorrect or poorly formulated assumptions.

Difficulties with the procedure include mapping from one to many from the set of behaviors to the set of metrics, i.e., relating a single performance consequence to several worth criteria. For example, if a robotic vehicle correctly senses and notes an enemy mine, this event could be relevant to the vehicle metrics for Intelligence and Effectiveness. Conversely, there can be a mapping from many to one, i.e., many performance criteria may be related to a single worth criteria. For example, behavior such as finding mines, avoiding rocks, and finding survivors, among others, contribute to the vehicle's Effectiveness.

Other difficulties include the existence of complex patterns of interaction among various aspects of performance and complex patterns of interdependence among subjective notions of worth (such as distinguishing among Intelligence, Effectiveness, and Efficiency; or among Command & Control, Communications, and Effectiveness. It can be difficult to distinguish between interactions among performance consequences (i.e., system behavior), which is a result of physical phenomena, and interdependence among worth criteria imposed by the analyst, which is a result of psychological phenomena. Nevertheless, an evaluation process that combines explicit subjectivity with objectivity is usually better than an evaluation process employing only implicit subjectivity. But "any assessment procedure, to generate comprehensible results, must stipulate very clearly whose point of view is being taken and whose values are to prevail," [28].

2.2 Worth

Underlying the evaluation procedure is the concept of worth, which may be defined as the "conscious perceptions held by an individual relating to his underlying feelings of preference, aversion, and indifference. This includes not only direct awareness of the feelings themselves, but also the entire range of cognitive elements supporting such feelings. Conscious rationalizations, justifications, and explanations would all be included in the meaning of worth," [28]. Worth is a function of an object, the situation in which the object is placed, and the person evaluating the object. Notions of worth are formulated by people observing external objects and they may be projected onto those objects; but worth remains in the subjective minds of the observers. Worth judgements are neither true nor false; they exist in-the minds of human beings.

Ideally, the metrics for intelligent systems should have certain properties. They should be *complete and exhaustive* in that all important performance objectives should be represented by the list of measures. They should be *mutually exclusive* in that no listed measure should encompass any other measure. The metrics should be restricted to performance objectives of the *highest importance*, derivable from lower criteria in a worth hierarchy. They should be relatively *independent* in that decision-makers should be willing to obtain additional satisfaction on one measure in exchange for reduced satisfaction on another measure at a rate relatively independent of the level of satisfaction already attained on each.

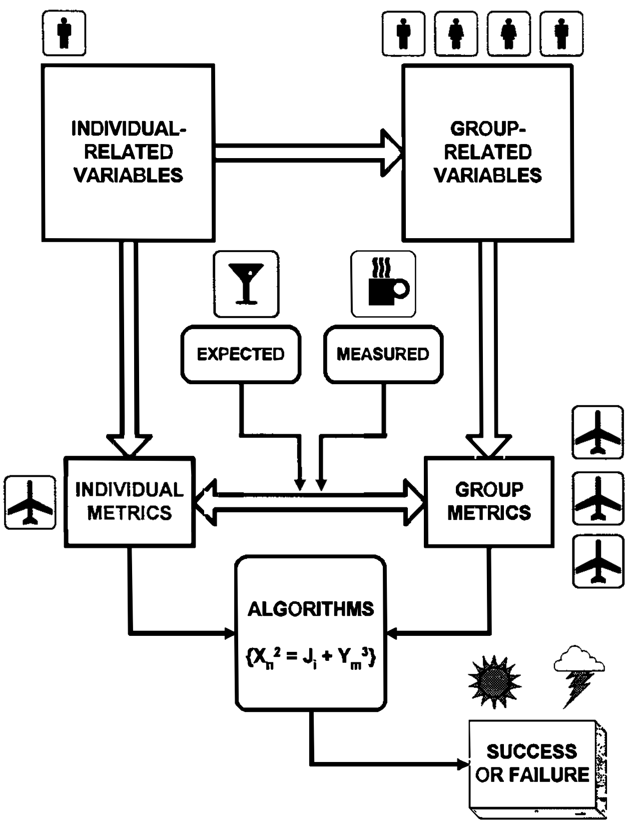
The example metrics selected herein for the intelligent systems intersect somewhat and are therefore *not completely mutually exclusive*, but their exclusivity is sufficient to provide a reasonable evaluation of system performance.

The lowest level criteria in a worth hierarchy should be represented by a simple performance measure. This connects the criteria hierarchy, which emanates from the subjective minds of the decision-makers, with the outer world of physical "reality." For example, the "Number of Targets Detected Per Unit Time" would be a lowest level worth criterion for the higher level criterion "Time Efficiency," which, in turn, would contribute to the evaluation for the higher level worth criterion "Mission Efficiency." The weighted worth scores may be aggregated to calculate an overall index of worth, i.e., an overall determination of success or failure for the intelligent system.

2.3 Variables

The variables and their relationships symbolically represent the operation of the intelligent system, in the context of the environment, in computer simulations or field exercises of missions for the intelligent system. **Figure 2** shows the relationships among the variables and the metrics. Some (although not all) of the system variables are relevant to the mission and group variables. Each of these sets of variables are aggregated, through the application of various algorithms, into metrics; these, in turn, are aggregated, through the application of more algorithms, into a scoring of success or failure. The values of the metrics, i.e., their quantification as a result of simulation or field exercises, determine the success or failure of the exercise of the system (against a priori criteria). In each case, the expected values (e.g., the martini glass in the figure) are compared with the measured values (e.g., the coffee cup in the figure) for the individual and group metrics and submetrics. Success or failure (e.g., of the mission) can depend on the individual, the group, or both.

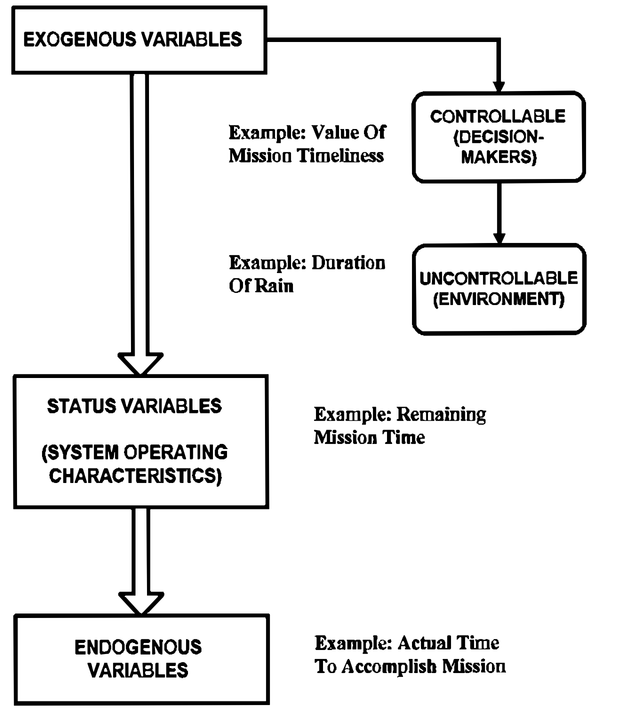
Figure 2. Variables And Metrics



A convenient taxonomy for the intelligent system variables is illustrated in **Figure 3**. Exogenous variables are independent, or input, variables which are generally predetermined and independent of the system. They act on the system but are not acted on by the system. Exogenous variables may be either controllable or non-controllable. Controllable (or instrumental) exogenous variables can be controlled or manipulated by the decision-makers of the system. Non-controllable exogenous variables are generated by the environment in which the system exists and behaves (and not by the system itself or its decision makers). For example, the value of "Mission Timeliness" is a controllable variable, while "Duration (Time) Of Rain" is a non-controllable variable. Non-controllable variables are associated with the individual level; there are none at the mission/group level.

The status variables describe the state of the system. They interact with both exogenous and endogenous variables according to the functional relationships of the system. The value of the status variable may depend on an exogenous or endogenous variable in a preceding time period; when the input is from a portion of a variable's own output from a previous period, a feedback loop exists. "Remaining Mission Time" and "Number Of Objectives Achieved" are examples of status variables.

Figure 3. Types Of Variables

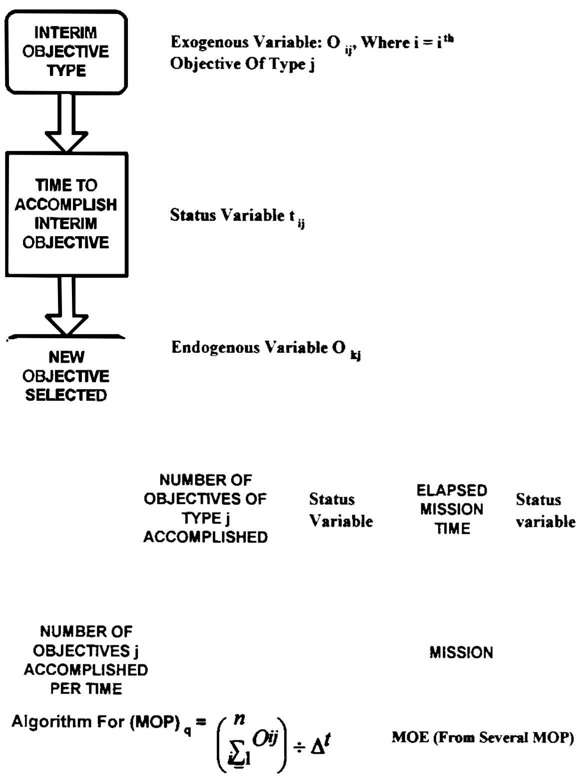


Endogenous variables are dependent, or output, variables of the system, generated from the interaction of the system's exogenous and status variables according to the system's operating characteristics. The "Actual Time To Accomplish A Mission" is an example of an endogenous variable.

Whether a particular variable is an exogenous, status or endogenous variable depends on the purpose or nature of the system's processes. For example, "Target Location" may be an exogenous variable if it is specified to the group a priori (as for a fixed target); it may be a status variable if, as a relative location, it is periodically updated as the group moves; and it may be an endogenous variable if it is computed by the group on the basis of sensor inputs.

An example of the use of the variables to derive metrics is given in **Figure 4**. Variables of different types are combined by using an algorithm to obtain a measure of performance: the exogenous variable "Interim Objective Type" (such as a rendezvous point); the status variable "Time Of Interim Objective Accomplishment;" endogenous variable "New Objective Selected" (by the leader vehicle); the status variable "Number Of Objectives" (of this type accomplished); and the status variable "Elapsed Mission Time." The MOP formed from these variables is the "Number Of Objectives Of Type j (such as rendezvous points) Accomplished (by the group) Per Unit Time." The algorithm in this example is simply the sum of the objectives accomplished divided by the mission time. This MOP, along with others (such as "Energy Expended Per Objective Accomplished"), might be combined into a top level metric called "Mission Efficiency."

Figure 4. Example: Variables Transformed To Metrics



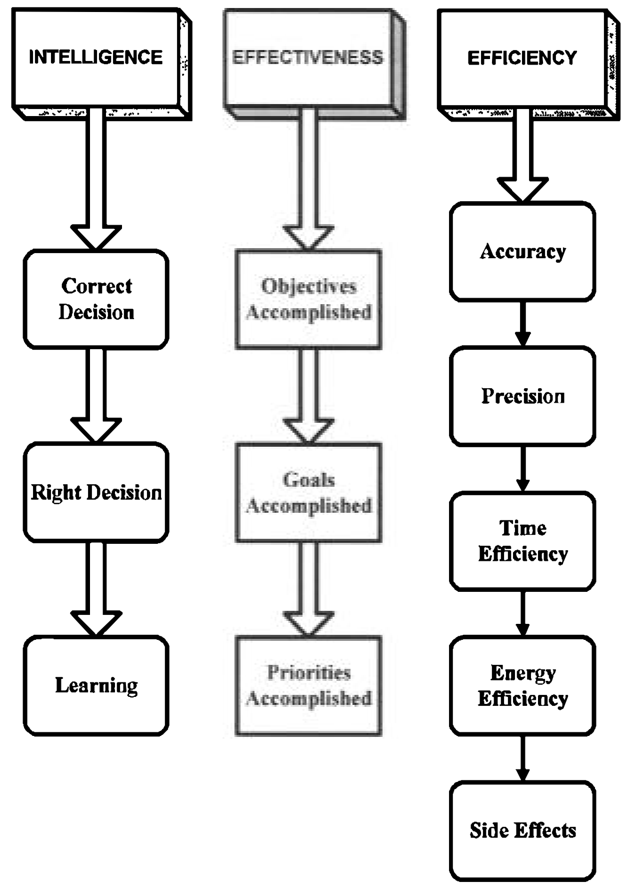
2.4 Metrics For Individuals And The Collective

We define six metrics for intelligent systems for this example. The metrics are not completely mutually exclusive. But they do emphasize three behavioral aspects of an autonomous intelligent system and three behavioral aspects of a group of such systems. Taken together, they provide a summary quantification of how successfully the individual and the aggregate - the collective - perform in the context of their environment and mission.

2.4.1 Individual Metrics And MOP

The three selected top-level metrics are: Intelligence, Efficiency, and Effectiveness, as shown in **Figure 5**.

Figure 5. Metrics And Submetrics For Individuals



Intelligence is defined here functionally as the ability of the system to make an appropriate choice - an appropriate decision. Because a value is the relative worth of a thing, the basis upon one makes a choice, intelligence is related to values; what is "appropriate" is situation-dependent. In the case of intelligent systems for military-type missions, appropriate choices are those that contribute to the success of the mission, or are perceived by the system to contribute to the success of the mission in the context of the information it possesses.

Information provided by sensors and processed by an intelligent control system can alter the intelligent machine's world model - and learning occurs. The ability to learn, based on experience, is one metric for intelligence. There are two kinds of acceptable decisions that the intelligent system can make: "correct" and "right." A correct decision is the optimum decision the system can make given a meta-view or complete knowledge (ground truth or the "god's-eye" view). The right decision is the optimum decision the vehicle can make given its "real" and limited knowledge. The intelligent machine (or a person) may do well in making correct decisions despite limited knowledge; this kind of decision-making is a metric that evaluates performance in an absolute frame of reference. It is difficult or impossible for mortals to acquire the "god's-eye" view in real life, but it is possible to have such a view in limited scenarios and to evaluate the performance of men or machines against such a standard.

In **Figure 5** the metric "Intelligence" is decomposed into the submetrics (or MOP): "Correct Decision," "Right Decision," and "Learning." "Correct Decision" evaluates the machine's intelligence against *absolute* performance standards. "Right Decision" evaluates the machine's intelligence against a *relative* standard which discounts the limitations of the machine's sensors and world model. The "Learning" MOP measures the ability of the vehicle to adapt to its environment, without necessarily making a decision that leads immediately to altered *external* behavior.

For example, an autonomous robotic vehicle might sense a terrain feature it that doesn't appear in the terrain map stored in its world model. Appropriate learning would occur if the vehicle were to alter its terrain map to include the feature; the vehicle need not have altered its path or motion in order to indicate learning - the change in the world model would be sufficient to indicate learning. If the vehicle were to select a path to its destination that complied with all of its mission criteria, but was then ambushed and destroyed by a hidden enemy about which it could not have known, the vehicle would have made a right decision in its path selection, but not a correct decision.

The metric "Effectiveness" in **Figure 5** is decomposed into the submetrics or MOP: "Objectives Accomplished," "Goals Accomplished," and "Priorities Accomplished." "Effectiveness" is the "bottom line" measure of merit, the measure of whether the mission goal and its interim objectives were achieved by the vehicle. Ordinarily, this might be the main metric, the one with the greatest importance. However, developmental or prototype systems may have, for example, various mechanical-type subsystems that are not of operational quality. It is not absolutely critical to the development of intelligent systems that prototype robotic vehicles accomplish its goals and objectives with overwhelming panache. The display of intelligence is more important in a Phase I effort than the success or failure of the mission - which may depend on the success or failure of a prosaic propulsion system. In the end, of course, with a fielded system, "Effectiveness" is a key metric. Ineffective intelligence is barren, in machines or people.

The tactical "Objectives Accomplished" is an MOP based on the intermediate objectives the intelligent machine is assigned to accomplish on its way to the ultimate mission goal, which accomplishment is accounted for in the MOP "Goals Accomplished." The final MOP for "Effectiveness" is the determination of the "Priorities Accomplished." The priorities are those set in the value-driven logic of the robotic platform, i.e., the relative importance of survival, energy conservation, timeliness, etc. The robotic platform may be able to accomplish most of its intermediate objectives, yet fail at its ultimate goal (just like people often do), or it may achieve its ultimate goal while failing at its intermediate objectives (e.g., getting the lucky break). Also, it may maintain or scramble its priorities while succeeding or failing at accomplishing its objectives and goal. The MOP for "Effectiveness" are thus sufficiently mutually exclusive to highlight different aspects of the robotic vehicle's behavioral and mission performance.

The final metric, that of "Efficiency," is the least important in a development program because a prototype platform's mechanical performance is likely to be inferior to that required for an operational platform. However, it is reasonable to account for this behavior in the testbed and include it in the final metric score. For an operational system, "Efficiency" becomes more important, but not usually as important as "Effectiveness."

"Efficiency" is a measure of how well the intelligent system performs while attempting to accomplish its objectives and goal, and how well it conserves resources. "Effectiveness" measures the ability to accomplish the objectives and goal assigned by the mission. The vehicle (like a person) may be extremely efficient and yet completely ineffective (such as working economically toward the wrong goal); or it may be inefficient, yet able to accomplish its objectives and goal. "Effectiveness" and "Efficiency" are not completely independent, but they are sufficiently different to characterize different aspects of an intelligent system.

There are four MOPs for the metric "Efficiency," as shown in **Figure 5**: "Accuracy," "Precision," "Time Efficiency," "Energy Efficiency," and "Unexpected Adverse Side Effects."

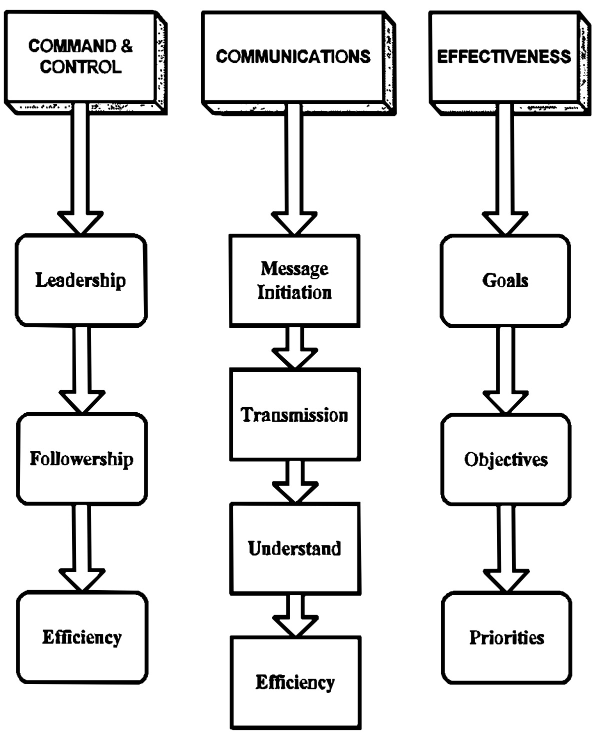
"Accuracy" refers to the robot's ability to achieve its desired states (position, speed, etc.) without significant systematic errors. "Precision" refers to the vehicle's ability to achieve its desired states without significant random errors. "Time Efficiency" measures the accomplishment of objectives and goal per unit time (such as the number of targets detected per minute, or the number of survivors retrieved per hour, the area searched per hour, etc.). "Energy Efficiency" likewise measures the accomplishment of objectives and goal per unit of energy expended (such as the number of mines detected per joule, etc.). The "Unexpected Adverse Side Effects" refer to adverse behavior displayed by the robot due to bugs, glitches, or errors in the vehicle. Such behavior may not prevent the vehicle from accomplishing its mission (or even detract much from its accuracy or precision), but it could reduce efficiency. For example, every 100 meters the robot might inexplicably stop for ten seconds; or it might mistake a wall for an entranceway and try to enter.

Accuracy and Precision are basic to efficient performance and should be weighted somewhat higher than Time and Energy Efficiency. Side Effects, while disturbing and potentially harmful to the success of the mission (or the continuation of a development program itself) is not of high importance in a Phase I development effort; the causes of eccentric behavior presumably can be found and corrected. The existence of peculiar vehicle bugs will become more worrisome as intelligent machines become operational.

2.4.2 Group Metrics And MOP

The three metrics selected for the group or mission level are: "Command and Control" (C^2), "Communications," and "Effectiveness," as shown in **Figure 6**.

Figure 6. Metrics And Submetrics For Groups



"Command and Control" (taken as a single measure) refer to the ability of the robots to exhibit cooperative behavior within a leadership structure. One major system attribute - intelligence - is measured from individual robot or platform behavior. The other major system attribute - cooperation - requires more than one platform for measurement; it is measured from the interactions of multiple intelligent systems. We assume an explicit means of achieving robotic group behavior for the applications of interest (e.g., leader-follower architecture), rather than implicit means (e.g., eusocial architecture).

The multiple systems can be designed to interact in many different ways, just as people in various societies and institutions organize themselves in different ways. In particular, the organizational forms needed to achieve organizational goals can range over a spectrum of types, from collegial to democratic to autocratic to - and so on. The organizational form selected, for example, may be military-autocratic where some robots are leaders and others subordinates, all in a hierarchy of authority and power. (Authority is the *right* to act while power is the *ability* to act).

In **Figure 6** the metric "Command and Control" is decomposed into the MOP: "Leadership," "Followership," and "C^2 Efficiency." While *cooperation* may seem too weak a characterization for the relationship between a military leader and his (its) subordinates, leadership always involves some form of cooperation from followers - even from those under duress.

The definition of "leadership," like that of "intelligence," is vague. Some of the definitions of leadership include [29]:

- "Leadership is the exercise of authority and the making of decisions," (Dubin, 1951);
- "Leadership is the initiation of acts that result in a consistent pattern of group interaction directed toward the solution of mutual problems," (Hemphill, 1954);
- "The leader is one who succeeds in getting others to follow him," (Crowly, 1928);
- "Leadership is the process of influencing group activities toward goal setting and goal achievement," (Stogdill, 1948);
- Leadership is "the ability to handle men so as to achieve the most with the least friction and greatest cooperation," (Munson, 1921);
- Leadership is "the process by which an agent induces a subordinate to behave in a desired manner," (Bennis, 1959);
- Leadership is "the activity of persuading people to cooperate in the achievement of a common objective," (Koontz and O'Donnell, 1955).

An ideal form of leadership might be to motivate others such that they perceive themselves to be **self-motivated**, an invisible, unobtrusive form of leadership. Then, there is the eusocial leaderless leadership, a commonality of purpose arising from the dynamics of group interactions, as exhibited by ants. Unlike human organizations, robotic systems might well be able to accomplish invisible or leaderless leadership.

Effective leadership can be measured by how well the leader's group performs its assigned functions in terms of group productivity and group satisfaction, although in the case of the robotic collective, group satisfaction is not a concern. In human organizations, the effective leader possesses power which originates from his position, from higher authority, and from his traits, abilities and behaviors. The followers of the leader also have traits, abilities and behaviors which contribute to the successful accomplishment of the mission. Between the leader and the followers are their relationships and the task structure. Technology impacts on the triad of leader, followers, and their relationship in various ways; communications technology, for example, can alter the leader's power or facilitate orders to subordinates.

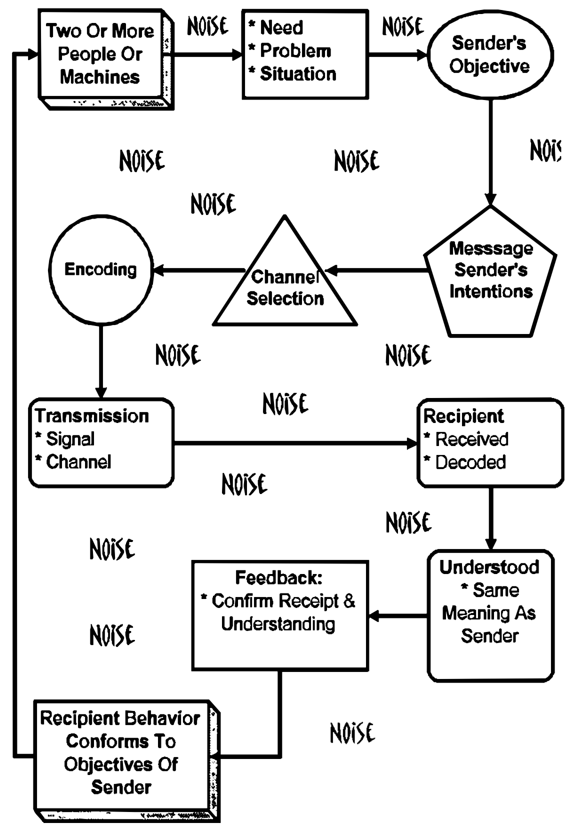
For a robotic collective, there will also be leadership potential (programmed algorithms), behavior (decisions based on value-driven logic), leader-follower relations (inter-vehicle protocols), and task structure (the degree of control and the tradeoff of centralization versus decentralization). The leader robot will take the initiative in making decisions, select tactics and maneuvers, and issue appropriate commands to the follower robot.

There can be no leader without a follower, and there can be no leadership without followership. So, for example, the followership of one robot vehicle will help define the leadership of another robotic vehicle. The subordinate vehicle will respond to commands appropriately, providing feedback to the command vehicle and behaving with restraint (an aspect of control). The subordinate vehicle will take command, transforming itself into the leader vehicle, when the latter cannot function properly because it has been damaged or destroyed. (In the case of two vehicles, the surviving follower becomes the "leader" in the nominal sense of performing the mission tasks of the leader vehicle without leading a subordinate).

The link between the leader and followers is achieved through communication. **Figure 7** illustrates the communication process. One of at least two or more people or machines perceive a need, problem, or situation that requires the transmission of information. The communication initiator - the sender - has an objective in sending the information. The sender formulates a message that contains the information reflecting the intentions of the sender. (Information, in a quantitative context, is a measure of one's freedom of choice when one selects a message from an available set. In this *entropic* view, the message that water is wet, to one who knows this fact, would not contain information. If information is related to choice, and values are the bases of choice, and intelligence is the ability to make appropriate choices, then information, values, and intelligence are related.) The sender selects a channel or medium over which to send the message, encodes the message into the appropriate language and format, and transmits the message over the channel. The recipient of the message receives and decodes it..

For communication to have taken place, the recipient must understand the message, i.e., must extract the information the sender intended. Ideally, the recipient provides feedback to the sender so that the latter knows that the information has been received and understood. Sometimes the feedback consists of the sender's observing subsequent behavior of the recipient that conforms to the objectives of the sender.

Figure 7. Communications



Any step of the communication process can be disrupted by noise. Noise may originate in the sender and disrupt the internal formulation or encoding of the message, or it may originate in the environment and disrupt the transmission of the message, or it may arise in the recipient and discombobulate the decoding or understanding of the message.

The communication process, as outlined **Figure 7**, is true for communication between people (using verbal, written, and other means) or robotic vehicles (using radio frequency, acoustic, optical, and other means).

The “Communication” metric is decomposed into four MOP: “Message Initiation”, “Message Transmission”, “Message Efficiency”, and “Message Understanding”. The MOP correspond to the communications process as outlined.

The initiation and understanding of messages are more important in a Phase I development effort than the performance of the transmission mechanism (radio frequency or acoustic), or the efficiency of the message protocol (length and number of messages needed to convey a quantity of information). Acceptable performance might consist of appropriate messages being initiated and a high percentage understood, given reception (but many messages might not be received to due noise or inadequacies in the transmission system).

The metric "Effectiveness" in **Figure 6** is decomposed into the MOP: "Objectives Accomplished," "Goals Accomplished," and "Priorities Accomplished." Effectiveness, as noted previously, is the *bottom line* metric, the measure of whether the mission goal and its interim objectives were achieved by the robotic vehicles. Ordinarily, this might be the main metric, the one with the greatest importance. However, a developmental robot may be a testbed with mechanical systems that are not of operational quality. As we mentioned previously, it is not absolutely critical to the success of a development program that the leader and follower vehicles accomplish their goals and objectives with overwhelming panache. The display of C^2 ability at the mission/group level (and intelligence at the individual level) is a more important accomplishment during development than the success or failure of the mission.

The tactical "Objectives Accomplished" is an MOP based on the intermediate objectives the robotic vehicles are assigned to accomplish on their way to the ultimate mission goal, which accomplishment is accounted for in the MOP "Goals Accomplished." The final MOP for "Effectiveness" is the determination of the "Priorities Accomplished." The priorities are those set in the value-driven logic (e.g., the relative importance of survival, energy conservation, timeliness, etc.).

The robotic vehicles may be able to accomplish most of their intermediate objectives, yet fail at their ultimate goal (just like people), or they may achieve their ultimate goal while failing at their intermediate objectives. Also, they may maintain or scramble their priorities while succeeding or failing at accomplishing their objectives and goal. The MOP for Effectiveness are thus sufficiently mutually exclusive to highlight different aspects of the leader-follower behavior and mission performance.

2.5 Metrics And The Analytic Hierarchy Process

There are multi-criteria decision-making techniques which can be used to define and weight metrics and evaluate alternative systems and technology for prospective intelligent robots. One such technique, the Analytic Hierarchy Process (AHP), is gaining popularity in the defense community (U.S. and Canada) for aiding in the evaluation of weapons systems, and there are more than 600 papers and books describing the theory and applications of the AHP. The mathematics underlying the AHP is largely matrix algebra wherein one solves for certain eigenvalues [30, 31, 32].

Making decisions about complex problems involving conflicting criteria and several alternatives is not a simple process. Psychological research has demonstrated that the human mind is limited in the number of items it can store in short-term memory. The AHP enables the decision-maker to transcend such limitations by visually structuring a complex problem in the form of a hierarchy. Each factor and alternative can be identified and evaluated with respect to other related factors. The AHP makes it possible to look at the elements of a problem in isolation: one element compared against another with respect to a single criterion. The decision process reduced to its simplest terms - pairwise comparisons. This ability to structure a complex problem, and then focus attention on individual components, improves decision-making. All judgements are synthesized into a unified whole in which the alternatives are clearly prioritized from best to worst.

For example, one might look at two robots and note (quantitatively) that the first weighs more than the second. In addition to observing this, we have an ability (subjectively) to say that the first robot is much more flexible (i.e., has an ability to perform more or varied functions) than the second, or just moderately more flexible, or that the flexibility of the two robots is the same. Or we might quantify the flexibility in terms of a measurable quantity (such as the number of defined functions performed), for example. A multiplicity of such pairwise comparisons of alternatives (or the use of objective data, where available), against various criteria, build a metric that can be used to make judgments or decisions that are more objective and rational than they would be otherwise.

We first performed this kind of analysis for determining robotic "IQ" for autonomous underwater vehicles in 1985 [32]. This work was updated for robotic ground vehicles in 2000 [33]. The results of this analysis is summarized below.

2.6 Example Analysis

As an example from longer lists [33], *exogenous* variables for *individual* robots include: coordinates (starting and final); maximum detection range (passive and active); terrain profile; object (size, speed, acceleration, coordinates; rendezvous coordinates; etc. Sample *status* variables include: vehicle speed (linear and angular); vehicle position; vehicle bearing; sensor status; power status; etc. Sample *endogenous* variables include: probability of bring detected (actively and passively); risk of known and unknown sensors along path; estimated path length; computed position of object sensed actively; computed object speed; etc.

Example *exogenous* variables for robot *groups* include: mission type; mission values; desired vehicle spacing; designated group leader; primary mission objective types (defenses, targets, vehicles, etc.); abort criteria; group clock standard; etc. Sample group *status* variables include: groups destroyed; vehicles per group destroyed; vehicles absent from rendezvous; elapsed/remaining mission time; etc. Sample group *endogenous* variables include: risk of active detection fo group; risk of passive detection for group; number of objects of each type sensed by group; best computed position of object sensed actively by group; etc.

For the AHP Goal to "Evaluate *Individual* Robot IQ," the values of the weights for the metrics and submetrics, previously described, were calculated with the following results:

* Intelligence = 0.54
- Correct Decision = 0.10
- Right Decision = 0.27
- Learning = 0.17

* Effectiveness = 0.30
- Objectives Accomplished = 0.15
- Goals Accomplished = 0.06
- Priorities Accomplished = 0.09

* Efficiency = 0.16
- Accuracy = 0.05
- Precision = 0.05
- Time Efficiency = 0.03
- Energy Efficiency = 0.02
- Side Effects = 0.01

For the AHP Goal to "Evaluate *Group* Robot IQ," the values of the weights for the metrics and submetrics, previously described, were calculated with the following results:

* Command & Control = 0.54
- Leadership = 0.23
- Followership = 0.23
- Efficiency = 0.08

* Communications = 0.16
- Message Initiation = 0.06
- Message Transmission = 0.03
- Message Understanding = 0.06
- Efficiency = 0.01

* Effectiveness = 0.3
- Goals Accomplished = 0.06
- Objectives Accomplished = 0.15
- Priorities Accomplished = 0.09

While there are many ways to evaluate the "IQ" of a robot and groups of robots, a simple (vector) method is to add the products of the values obtained for the individual and group metrics and their associated weights:

[1] Total Score = $\sum_i W_i M_i$

Where W = i^{th} Weight
M = i^{th} Measure (Score)

The scores of each metric are obtained from measuring the submetrics or MOPs in a series of experiments, in a simulation or in the field. Each individual and group metric requires a defined process for obtaining its score, which is then aggregated into the Total Score (or "IQ"). There are many possible approaches or algorithms, an examples are given in [33]. For example, to evaluate the group Communications metric one might define:

[2] $SC = \sum_{i=1}^{3} W_i R_i + W_4 E$

Where:

* SC = Score For Communications
* R_1 = NMI/TMI = Message Initiation Ratio
* R_2 = NESR/TMI = Transmission Ratio
* R_3 = NMU/NESR = Understanding Ratio
* NMI = Number Of Right Messages Initiated
* TMI = Total Number Of Messages Initiated
* NESR = No. Messages Actually Encoded, Sent, And Received
* NMU = No. Of Messages Rightly Understood By Recipient
* W_i = Weight of i^{th} MOP (As Previously Calculated)
* E = Evaluation Of Message Lengths And Quantity Compared With What Would Be Right: $(0 \leq E \leq 1)$

Example steps to measure the MOP associated with group Communication include:

Step 1: Store the time of initiation of messages (i.e., a new plan of a robot to send a message to another robot), the contents of the messages, the time of transmission of the messages, the time of reception of the messages, and the contents of the messages as received by the receiving robot.

Step 2: The analyst, after the mission, calculates the Message Initiation Ratio, Transmission Ratio, and Understanding Ratio. The analyst judges the rightness of the message contents, as well as the rightness of the understanding of the messages on the part of the receiving robot, based on the robot's subsequent behavior. The analyst also judges the rightness of the message lengths and quantity (too much or too few) of messages and scores this as previously described.

Step 3: The analyst weights and combines the scores of the four MOP associated with group Communications to calculate the Communications Score, and weights and combines this score with the other weighted metric scores to obtain a final value for the group "IQ."

Another example is a method for scoring the Effectiveness metric for the individual robot. To score the accomplishment of the tactical objectives and goal of a mission, the human evaluator notes the number of objects (e.g., survivors in an urban search and rescue operation) to be sensed or acted upon (e.g., located, given water or oxygen, carried to safety) by the robot and divides by the total number of such objects in the scenario. A similar ratio is taken for the number of positions (rendezvous locations, assigned reconnaissance positions, etc.) the vehicle should have visited. The Priority MOP is evaluated by determine whether the priorities in the value-driven logic were followed as assigned, or modified according to the rules, through the mission. For example, a score (e.g., 0 to 4) can be assigned to each priority, then they are summed and averaged. For Effectiveness we then have:

$$[3]\quad R(O_{ij}) = (\sum_{j}^{m} \sum_{i}^{n} O_{ij}) / O_T$$

$$[4]\quad R(P_i) = (\sum_{i}^{m} P_i) / P_T$$

$$[5]\quad S(\mathrm{Pr}) = \sum_{i=1}^{4} \mathrm{Pr}_i / 4$$

Where $[0 \leq \mathrm{Pr}_i \leq 4]$

Where:
* $R(O_{ij})$ = Object ratio (for goal or objectives)
* $R(P_i)$ = Position Ratio (for goal or objectives)
* $S(Pr)$ = Priority Sum Average
* O_{ij} = The i^{th} Object of Type j (For example, j=1=survivor; j=2=mine; j=3=areas to be avoided; etc.)
* P_i = The i^{th} position (goal or objective) Visited
* m = Total Types of Objects Sensed or Acted Upon (Or Total Position Visited)
* n = Total Objects Of Each Type Sensed Or Acted Upon
* O_T = Total Number Of Objects Robot Should Interact With To Achieve Goal Or Objectives
* P_T = Total Number Of Positions (Goal Or Objectives)
* Pr_1 = Stealth
* Pr_2 = Survival
* Pr_3 = Timeliness
* Pr_4 = Energy

The steps to measuring robot "Effectiveness" are:

Step 1: Specify the tactical plan for the mission. In an urban search and rescue mission, for example, this might be to: Search for a specified object or person; perform Reconnaissance (to search for entrances or signs of life); perform Surveillance (in a specified region); Map (a specified region); Retrieve a person, etc. If the mission goal for a group of two robots were to locate and retrieve survivors from within a room on an upper floor, the mission goal of one of the robots might be to locate a path to the upper floor by searching a lower floor. The mission-level goal consists, for example, of a state-graph defining a sequence of potential commands that the mission executor will issue to the group level planner. Store the robot's mission goal as specified at the start of the mission, and any changes of the goal made during the mission, with the time of the changes.

Step 2: Store the robot's input tactical commands for decomposed intermediate objectives (if they are changed during the mission, store the changes along with the times of the changes), then store the changes in the state-graph which indicate that a robot's input command has been accomplished by the robot, and note the time of the accomplishment.

Step 3: Determines whether the robot has substantially accomplished its mission goal. Calculate the score quantitatively e.g., using an Object Ratio or Position Ratio (for example, the ratio of entrances to a collapsed building located to the total number of entrances in the building) or qualitatively (assign a score to the mission).

Step 4: The Object Ratio and Position Ratio are used by the analyst to calculate the Objective Score, summing the number of objects or positions that the robot interacted with in the accomplishment of its intermediate objectives and dividing by the total number of such objects or positions with which it should have interacted (according to ground-truth).

Step 5: The values used in the value-driven route planner, such as for stealth, survival, timeliness, and energy, should be stored for retrieval by the analyst. At the conclusion of the mission, the analyst calculates the Priority Sum Average by evaluating the behavior the vehicle, assigning a scores, and taking an average.

Step 6: The analyst weights and combines the scores of the three MOP associated with vehicle Effectiveness (i.e., Goals, Objectives, and Priorities) and calculates a total score for "Effectiveness."

3.0 Acknowledgments

We gratefully acknowledge that this work was funded in part by Contract 43NANB009812 from NIST.

4.0 References

[1] Conant, Roger, ed. *Mechanisms Of Intelligence: Ross Ashby's Writings On Cybernetics*. Seaside, CA: Interscience Publishing; 1981, p. 178.
[2] *Glossary On Cybernetics And Systems Theory*, American Society For Cybernetics; 1984, pp. 21-22.
[3] Gevarter, William B. *Intelligent Machines*. Englewood Cliffs, NJ: Prentice-Hall, Inc.; 1985, p. 229.
[4] *The Economist*; 17 October 1987, p. 13.
[5] Gregory, Richard L. *The Oxford Companion To The Mind*. Oxford: Oxford University Press, 1987, p. 410.
[6] Chen, Minder, Yihwa Irene Liou, and E. Sue Weber.

Developing Intelligent Organizations: A Context-Based Approach to Individual and Organizational Effectiveness. *Journal of Organizational Computing* 2(2): pp.181-202, 1992.
[7] McCarthy, J., and P. Hayes. Some Philosophical Problems From The Standpoint Of Artificial Intelligence. In *Readings In Artificial Intelligence*, Bonnie Lynn Webber, Nils J. Nilsson, eds. Palo Alto, CA: Tioga Publishing Co 1981, p. 432.
[8] Fischler, Martin A., and Oscar Firschein. *Intelligence: The Eye, The Brain, And The Computer.* Addison-Wesley Publishing Co., Inc.; 1987, p. 4.
[9] Schrage, Michael. Book Renounces The Early Promise Of Artificial Intelligence. *Washington Post*, 1 June 1987, p. 17.
[10] Wooldridge, Dean E. *Mechanical Man: The Physical Basis Of Intelligent Life.* New York: McGraw-Hill Book Co.; 1968, p. 128.
[11] Ashby, W. Ross. *Design For A Brain.* Chapman & Hall; 1976.
[12] Doyle, Jon. *A Model For Deliberation, Action And Introspection,* MIT Artificial Intelligence Laboratory TR-581 May 1980, pp. 15-16.
[13] Barr, Avron, and Edward A. Feigenbaum. *The Handbook Of Artificial Intelligence.* Stanford, CA: Heuristic Press; 1981, p. 147.
[14] Rorty, Richard. *Philosophy And The Mirror Of Nature.* Princeton University Press; 1979, p. 125.
[15] Sowa, J.F. *Conceptual Structures: Information Processing In Mind And Machine.* Addison-Wesley Publishing Co.; 1984, p. 2.
[16] Gregory, op. cit., p. 149.
[17] Gevarter, op. cit., p. 225.
[18] Fischler, op. cit., p. 383.
[19] Minsky, Marvin, and Seymour Papert. *Perceptons.* MIT Press; 1969.
[20] Pugh, George E. *The Biological Origin Of Human Values.* Basic Books Inc.; 1977.
[21] Gregory, op. cit., p. 383.
[22] Koza, John R. *Genetic Programming.* MIT Press; 1992, p. 14.
[23] Stonier, Tom. *Beyond Information: The Natural History of Intelligence.* Springer-Verlag; 1992, p. 70.
[24] Stonier, op. cit., p. 72.
[25] Stonier, op. cit., p. 85-105.
[26] Arkin, Ronald and George Bekey, *Robot Colonies*, Kluwer Academic Publishers, 1997, reprinted from the special issue of the journal of *Autonomous Robots*, Vol. 4, No. 1, March 1997.
[27] Cao, Uny Y., Fukunaga, Alex S., and Andrew B. Kahng, *Cooperative Mobile Robotics: Antecedents And Direction*, in Arkin, Ronald and George Bekey, *Robot Colonies*, Kluwer Academic Publishers, 1997, pp. 7- 27.
[28] Miller, James R., *Professional Decision-Making: A Procedure For Evaluating Complex Alternatives*, Praeger Publishers Inc., 1970.
[29] Finkelstein, Robert, "Combat Robotics: Implications For Leadership," *Unmanned Systems*, Winter 1987, pp. 9-10.
[30] Saaty, Thomas L., *Decision Making For Leaders: The Analytical Hierarchy Process for Decisions in a Complex World*, Wadsworth, Inc., 1982.
[31] Saaty, Thomas L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill Book Co. 1980.
[32] Saaty, Thomas L., A Scaling Method for Priorities in Hierarchical Structures, *Journal of Mathematical Psychology*, 15, pp. 234-281 (1977).
[32] Finkelstein, Robert, *Measures Of Performance And Effectiveness For The MAUVE System: MOE Report Number 5*; Robotic Technology Inc., Item Number 0002AE, Contract No. 50SBNB7C4549, National Institute Of Standards And Technology, June 1987.
[33] Finkelstein, Robert, *Metrics: Evaluating The Performance Of Intelligent Systems*; Robotic Technology Inc., No. 43NANB009812, National Institute Of Standards And Technology, June 1987.

5.0 Author Biography

Dr. Robert Finkelstein is an Associate Professor and Director of the MBA Program at the University of Management and Technology and an Adjunct Associate Professor at the University of Maryland. He is President of Robotic Technology Inc. and Chairman of Decide-Now.Com, Inc. He received a DBA in Systems and Cybernetics from the George Washington University (GWU) in 1995; Ap.Sci. in Operations Research from GWU in 1977; M.S. in Operations Research from GWU in 1974; M.S. in Physics from the University of Massachusetts in 1966; B.A. in Physics from Temple University in 1964.

Machine Intelligence Ranking

Paul P. Wang
Duke University, BOX 90291
Department of Electrical and Computer Engineering
Durham, North Carolina, 27708-0291, USA

ABSTRACT

This talk addresses a number of issues which were inspired by the draft of a document on Metric for Intelligence of Constructed Systems. The constructed systems here literally mean an autonomous control system. It is important to note the opinions expressed in this talk reflect the thoughts of the author and they do not reflect on any institution, organization or professional society.

There are six issues being raised in this talk. The first issue deals with the discussion on the role NIST should play. The institution of NIST was chartered to serve American citizens to improve their well being and the noble goal of pursuing a life of happiness. One of the most important tasks is to measure, standardize, and rank the engineering systems and the advancement of the technology objectively. The autonomous constructed systems were singled out with a high profile to reflect their importance. Are there any other man-made systems which are equally or more important?

Second issue has to do with measuring intelligence. We are measuring intelligence because technology embraces intelligence giving us a superior and high performance system. On the other hand, it is not NIST's mission to do all that because it is there! The fundamental issue, however, is to serve the citizens better via improved technology which requires intelligence. The definition of intelligence, however, is no simple matter, as well as the definition of serving citizens. Both cover a wide spectrum of needs and desirable things other than autonomous systems of which intelligence so happen needed to be put in the center of the stage.

The third issue to be raised is the definition of "machine intelligence" and how to measure it? Since the definition of human intelligence is complex and difficult, the definition of machine intelligence is even more difficult!

The fourth issue has to do with the performance evaluation of engineering systems. This issue deals with value judgement. The debate by the citizens among all walks of life and society as a whole must be carried out in order to establish value judgement as a benchmark for measurement, testing, and evaluations.

This brings us to the issue of testing and measuring. The central issue is how are we to conduct the machine intelligence test? It is not a simple matter because we have not yet settled the definition of machine intelligence!

Equally important is the issue of understanding the crux of our present technology and forecasting of future technology. The reason is due to the fact that there is absolutely no unique way to realize a high performance system. Here we are talking about a federal institution to set the standard to evaluate and rank a high performance system.

Generally speaking, the position this paper takes is that some of the issues raised in white papers are over simplified. Some of the long term frame works have not been covered adequately. If one believes in the basic assumptions, hypotheses set by the white paper and willing to live with all the constraints already being laid out, then this paper has no validity. The feeling of this author is that the constraints dealing with intelligent machines are overly constrained and a liberation effort hence is needed.

The main concerns are: the basic charter of the institution is unclear, the science on intelligence is too complex, the need of application areas is too complex, and the technologies available are too uncertain to reach a consensus.

With these constraints, I must say that the white paper is truly an outstanding document full of creativity, imagination, and innovative ideas. Congratulations to Alex Meystel and Jim Albus.

Survivability and Competence as Measures of Intelligent Systems

Reid Simmons
Robotics Institute
Carnegie Mellon University

While the workshop is appropriately named "Measuring the Performance of Intelligent Systems", there may be come confusion that the goal is actually about measuring the i*ntelligence of systems.* While measuring performance is a worthy, albeit difficult goal, I believe that trying to measure intelligence itself is misplaced. To me, it seems pointless to debate whether, for instance, playing chess exhibits more "intelligence" than exploring Mars, or whether using speech is inherently more intelligent than doing object recognition. From both pragmatic and philosophical viewpoints, the more that we can make it clear that we are interested in *performance*, rather than intelligence, *per se*, the better off we will be.

So, what criteria are to be used for measuring the performance of intelligent systems? I think that the two most important characteristics are survivability and competence. By *survivability,* I mean the ability of a system to cope with diversity in the environment, as well as internal faults (hardware and software). By *competence*, I mean the ability of a system to successfully perform tasks. Both survivability and competence can be measured either empirically or formally. Empirically, survivability can be measured by carefully controlling environmental inputs and by modifying the internal state of the system (such as by deliberately causing hardware faults). Formally, with the right model one can quantify the range of environmental conditions and internal states that can be handled successfully. Similarly, one can measure competence either empirically or formally by controlling for the range of tasks and the environments under which those tasks are to be performed.

This, of course, begs the question as to how to set up the experiments in an unbiased and controlled fashion, and how to model tasks and environments so that formal evaluations are possible. Unfortunately, I do not have good answers for those questions, at this time (although we are working on it!). The problem is that most intelligent systems exhibit chaotic behavior - small deviations in input conditions lead to wide deviation of behavior (of course, many intelligent systems are also chaotic in the colloquial sense, but that is another matter...). Thus, it is very difficult to set up "the same" conditions to test different systems. One can never be sure if the results are due to actual differences between the systems themselves, or due to small differences in the environments. While simulation can be used to perform standardized experiments, simulators have the disadvantage that they tend to be rather simple models of reality, and so may not capture the essence of what makes survivability and competence difficult.

What about things like robot competitions and Turing tests? I am all for them, but not as quantitative measures of performance, since they suffer from the problem of variability, as described above. The reason that they are valuable is that they come *close* to standardizing tasks and environments in realistic settings, and so can be used by *developers* of intelligent systems to gauge progress, in qualitative ways, against the state of the art. While it is dangerous to use the results of such competitions to conclude anything about one system vs. another (especially one technology vs. another, such as neural nets vs. expert systems), competitions are useful as a type of "bread-boarding" exercise.

Finally, an important aspect of intelligence is *adaptability.* The question is whether adaptability should (or can) be measured independently from survivability and competence. I would argue that adaptability is merely one way of increasing a system's survivability and competence, and thus should not be considered independently. While it may turn out to be true that adaptable systems are generally more survivable and competent, it seems clear to me that this is a hypothesis that needs to be demonstrated empirically, or proved formally. In the absence of such proof, it seems to make little sense to measure adaptability in isolation.

In summary, survivability and competence are two critically important characteristics of intelligent systems. While it is possible to devise ways of measuring both, in a rigorous fashion, it is difficult due to the fact that autonomous systems interacting with complex environments tend to be chaotic. But, that fact should not lessen our resolve to try and measure performance - it only serves to make us aware of the limitations and difficulties of the enterprise.

Two measures for the "intelligence" of human-interactive robots in contests and in the real world: expressiveness and perceptiveness

Illah R. Nourbakhsh
The Robotics Institute
Carnegie Mellon University

Practical measures of intelligence are generally predicated on a social-anthropocentric view of intelligence. This is hardly surprising, but is undesirable because it results in intelligence testing procedures that are uninformative when the subject is not human. For example, the classical Turing Test measures machine intelligence using the yardstick of human social dialogue, in written form, as its gold standard. The problem is that such methodology is implicitly pass fail. Rather than providing a relative measure for machines that are clearly inferior to humans at social human interaction, this test simply fails all such machines until and unless some superior machine simply passes. In airness, it is possible to mitigate this to a small degree by narrowing the content area of the test.

Nevertheless, the Turing Test as applied to the mobile robot system suffers generally the same fate. One can imagine, for instance, a robot Turing Test in which the human teleoperated robot is compared in performance to an autonomous robot in tasks such as navigation, manipulation and robot-human interaction. But the robot will continue to suffer because its raw percepts and raw effectors are not comparable to that of a human. The solution, to force the teleoperating human to use the same percepts as the robot itself uses, results in a robot that whether teleoperated or not is disappointingly unintelligent even when it successfully passes such a robot Turing Test. The problem, then, is that a robot's potential for interaction imposes an upper bound on its potential for intelligence.

Based on this premise, I will propose in my talk that the form of intelligence about which we care most in the case of autonomous robots is interaction.

I will present a methodology for measuring the potential of a robot to engage in rich interaction, thereby establishing a behavioral and analytical way of measuring intelligence without reverting to a direct anthropocentric pass fail test. I will define the concepts of expressiveness and perceptiveness, which together place both upper bounds and lower bounds on interactivity and thereby intelligence. Expressiveness is a measure of the output richness of an electromechanical system. One can quantify expressiveness in terms of the average effectory branching factor of an agent in its observable output space.

Perceptiveness is a measure of the fidelity of an electromechanical system's effective mapping from environmental change to output. This too can be quantified by computing the set of possible output trajectories of an agent in its perceptual workspace. These two measures prove to be particularly useful because they contain no bias with respect to behavior-based and model-based robot architectures. After defining expressiveness and perceptiveness, I provide some quantitative results comparing the expressiveness and perceptiveness of a simple unicellular organism, the dinoflagellate, to that of several popular mobile robots. These quantitative results demonstrate that from the perspective of interactivity mobile robots have a long way to go before challenging human intelligence.